6DoF Cooperative Localization for Mutually Observing Robots

Olivier Dugas, Philippe Giguere and Ioannis Rekleitis

Abstract The solution of cooperative localization is of particular importance to teams of aerial or underwater robots operating in areas devoid of landmarks. The problem becomes harder if the localization system must be low-cost and lightweight enough that only consumer-grade cameras can be used. This paper presents an analytical solution to the six degrees of freedom cooperative localization problem using bearing only measurements. Given two mutually observing robots, each one equipped with a camera and two markers, and given that they each take a picture at the same moment, we can recover the coordinate transformation that expresses the pose of one robot in the frame of reference of the other. The novelty of our approach is the use of two pairs of bearing measurements for the pose estimation instead of using both bearing and range measurements. The accuracy of the results is verified in extensive simulations and in experiments with real hardware. At 6.5 *m* distance, position was estimated with a mean error between 0.021 *m* and 0.025 *m* and orientation was recovered with a mean error between 0.019 *rad* and 0.037 *rad*. This makes our solution particularly well suited for deployment on fleets of inexpensive robots moving in 6 DoF such as blimps.

1 Introduction

In GPS-denied environment, such as underground, underwater, or indoors, especially when features are sparse, the problem of localizing a team of robots can be addressed by the employment of the Cooperative Localization (CL) [22] methodology. In CL, robots use observations of each other in order to calculate their relative pose (position and orientation) without resorting to measuring features in the environment. The vast majority of research on CL has focused on the 2D case, with the

Olivier Dugas and Philippe Giguere
Département d'informatique et de génie logiciel, Université Laval, 3976-1065 Avenue de la Médecine, Québec, QC, Canada G1V 0A6. e-mail: philippe.giguere@ift.ulaval.ca Olivier.Dugas.1@ulaval.ca

Ioannis Rekleitis
School of Computer Science, McGill University, 318-3480 University Street, Montréal, QC, Canada H3A 2A7 e-mail: yiannis@cim.mcgill.ca

advent of more advanced mobile robots such as aerial, underwater or even rough terrain (outdoor) robots, the problem of estimating the relative pose in 3D is gaining popularity. Even when GPS is available, its accuracy in estimating the pose is insufficient for precise maneuvering, as compared to CL solutions.

More formally, for a team of robots, Cooperative Localization can be defined as follows: given a set of inter-robot measurements, estimate the relative pose of each robot with respect to the other team members. If one robot is stationary, then the pose of the other team members can be defined with respect to the stationary robot.

In this paper, we present an analytical solution for the 3D transformation that describes the relative pose between two robots, using a limited number of bearing measurements. We show how one can use the angle measurements obtained by cameras to estimate the relative 3D position and orientation of mutually observing robots. In the following section, an overview of related work is provided. Section 3 describes the 3D problem we are addressing, along with the proposed analytical solution. Section 4 presents the tracking experiment setup and Section 5 shows the results. Thereafter, Section 6 investigates the effects of a non-collinear camera-markers setup on the solution. Finally, we conclude with future directions for this work, and lessons learned.

2 Previous Work

Estimating the relative pose of a group of robots was presented as early as 1994 [18]. Early work focused on 2D pose estimation for localization [17] and mapping [20] using either a Kalman [23] or a Particle [21] filter. An interesting variation appears when the measurements do not come from identifiable robots (anonymous) [10]. More recently, work on using bearing only measurements produced an accurate solution for the 2D case [12]. It is worth mentioning that using bearing only measurements from a camera is an inexpensive solution which scales to large number of robots.

Employing Cooperative Localization to derive the 6-DoF relative pose measurements has gained popularity lately. In an application to the underwater domain, Bahr et. al. used a 2D approximation [1] by maintaining the same depth. The anonymous measurement solution to CL was extended in [3] using relative bearings and inertial data. The main contribution on this work is the data association, which is achieved by the additional ego-motion sensor data. Zhou and Roumeliotis presented a extensive list of solutions for Cooperative Localization in 3D using combinations of range and bearing measurements in conjunction with ego-motion measurements [28, 29, 30]. Contrary to our solution, a single bearing measurement is recorded each time from each robot, making the use of proprioceptive sensors a requirement. Cristofaro and Martinelli [4] presented an observability analysis for a state estimation system using bearing measurements and inertial data. In their formulation, the inertial data is necessary to recover the full pose. The reader will note that the observability analysis can be used to improve the EKF-based estimators in CL [13].

The accuracy of vision-based systems for distance measurements is limited, mainly due to discretization errors. However, many techniques have been developed to improve the precision using bearing measurements, since cameras are good protractors. For instance, bearing measurements were used in [27] to estimate the configuration parameters of a continuum robot. Also, implicit localization methods were presented with bearing-only measurements in [15] and [7] showed that relative measurements localization can be an NP-hard problem. Polynomial solvers were developed and optimized for solving minimal geometry problems [19]. In [6] a numerical solution was proposed for estimating the 3D relative pose using bearing only measurements. The accuracy of the reported results are lower than the ones obtained from the analytical solution proposed in this paper.

One of the oldest minimal geometry problem in computer vision is known as the Perspective-n-Point (PnP) problem. It is defined as the determination of the absolute pose of a camera given a set of correspondences between 3D points and their projections on the 2D image taken by the camera. Closed-form solutions have been proposed to the minimal absolute pose problems with known vertical direction [16] and also to the three points absolute pose problem [9].

Other vision-based approaches used different sensors to provide the bearing of an observed robot. Omni-directional cameras were used in [14] and [11]. Also, bearing and distance estimations were provided using active lighting with stereo vision in [5] or by combining vision with range finders in [2]. Relative bearing and/or range measurements has been employed in a constraint optimization framework [26]. Our solution, however, differentiates itself by the fact that only three to four angles and no distance estimation are needed to estimate the pose of an observed robot.

3 Solution to the 3D Problem

The 3D cooperative localization problem using bearing only measurements can be described as follows. A pair of robots, $Robot_A$ and $Robot_B$, are free to move in all six DoF, position [x,y,z] and attitude [pitch, roll, yaw]. Each robot $i \in \{A,B\}$ is equipped with a single camera and two visual markers, R_i and L_i. In this paper, we will use the following notation. Please note that, a 2D point in an image is expressed by a lower case bold $\mathbf{p}$, the subscript indicates which marker (L or R) is viewed, and the superscript on left the camera, e.g.: ${}^{A}\mathbf{p}_R$ is the right marker of robot B as seen by camera A. 3D points are represented as uppercase letter and 3D vectors are represented with an *arrow*, for example $\overrightarrow{R_i}$. A matrix is shown as a *bold* uppercase letter, e.g.: $\mathbf{R}$. Finally, unitary rotation axis have *hats* (e.g.: $\hat{e}_1$) and scalar elements do not have any specific notation.

The two visual markers are placed at a distance d from each other. The camera is rigidly attached on the respective robot so its center of projection C_i is collinear with these two markers and between them, at a distance d_{Ri} from R_i and at a distance d_{Li} from L_i. Each local frame of reference F_i is chosen to originate at the center of projection C_i, with the $\overrightarrow{z}$ axis being the optical axis and the $\overrightarrow{y}$ axis pointing upward. In a local frame of reference, the camera's center of projection C_i and the

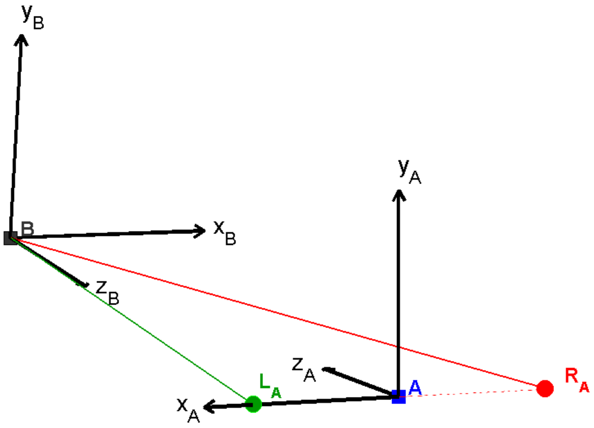

Fig. 1 The relative localization problem in 3D, for the two robots A and B operating in 6 DoF. In this depiction, we only show the markers on robot A. The red and green lines represent the ray between these markers and the center of projection of $Camera_B$, located at the origin of F_B.

two markers are thus located at:

$$C_i = [0\ 0\ 0], \qquad R_i = [-d_{Ri}\ 0\ 0], \qquad L_i = [d_{Li}\ 0\ 0].$$

Please note that the x coordinate for the right marker R_i is negative, as per the orientation of the frame of reference F_i; see Fig. 1. Finally, we assume that at all time, each camera can see the other robot and its two landmarks.

The relative pose between $Robot_A$ and $Robot_B$ is calculated by using two images recorded at the same time. Let the picture taken by $Robot_A$ be I_A, and the picture taken by $Robot_B$ be I_B. Each robot i, along with its two markers R_i and L_i, is present in the image acquired by the other robot, i.e. R_B and L_B appear in I_A and vice versa. This situation is illustrated in Fig. 1. The relative orientation and position between F_A and F_B will be expressed as a translation matrix ${}^A_B\mathbf{T}$ and a rotation matrix ${}^A_B\mathbf{R}$. The information available to find this transformation is:

- the position of the markers $R_i = [-d_{Ri}\ 0\ 0]$ and $L_i = [d_{Li}\ 0\ 0]$ within their robot's frame of reference F_i;
- the camera calibration internal parameters;
- sub-pixel location ${}^A\mathbf{p}_R$ and ${}^A\mathbf{p}_L$ in image I_A of R_B and L_B, respectively;
- sub-pixel location ${}^B\mathbf{p}_R$ and ${}^B\mathbf{p}_L$ in image I_B of R_A and L_A, respectively.

From the above information, we can also infer the approximate position of the other robot's camera in an image, given that it is located by construction between the two markers:

- sub-pixel location in image I_A of C_B as

$$ {}^A\mathbf{p}_B \approx {}^A\mathbf{p}_R + \left(\frac{d_{RB}}{d_{LB}+d_{RB}}\right)({}^A\mathbf{p}_L - {}^A\mathbf{p}_R); \qquad (1)$$

- and sub-pixel location in image I_B of C_A:

$$^{B}\mathbf{p}_A \approx {}^{B}\mathbf{p}_R + \left(\frac{d_{RA}}{d_{LA}+d_{RA}}\right)({}^{B}\mathbf{p}_L - {}^{B}\mathbf{p}_R). \tag{2}$$

This approximation holds when the robots are sufficiently far apart ($l \gg d$) or when the robots are perfectly facing each other. Please note that ${}^{i}\mathbf{p}_X$ indicates the position of point X (camera, or marker R or L), in the frame of reference of i.

3.1 Reduction to 2D problem

Let us first consider the plane defined in 3D space by the three collinear points (by physical construction) R_A, L_A, C_A on robot A and the center of the camera on robot B, point C_B. Note that the origins of the two frames of reference F_A and F_B are also part of this plane, since the origin of frame F_i is defined as the center of projection C_i of a camera. This is the key to our solution, as it allows to reduce part of the 3D calculation into a 2D problem, the solution of the reduced problem being identical to the one presented in [12] and depicted in Fig. 2. In particular, the 2D solution estimates the distance between the two cameras $l = |C_AC_B|$.

As described in [12], the estimation of the relative position requires the measurement of two angles, α and β and knowing the distance d between the markers L_A and R_A. In the current formulation, the angle $\widehat{L_AC_BR_A} = \alpha$ is defined within that plane and is computed straightforward by creating the 3D vectors going from the origin to the image plane of I_B

$$^{B}\overrightarrow{P}_R = [{}^{B}p_{Ru} \quad {}^{B}p_{Rv} \quad f_B] \tag{3}$$

and

$$^{B}\overrightarrow{P}_L = [{}^{B}p_{Lu} \quad {}^{B}p_{Lv} \quad f_B] \tag{4}$$

where f_B is the focal length of $Camera_B$. Note that the subscripts u, v indicate the horizontal and vertical pixel location in an image. The angle α is then the angle between these two vectors, computed as:

$$\alpha = acos(\frac{{}^{B}\overrightarrow{P}_L . {}^{B}\overrightarrow{P}_R}{|{}^{B}\overrightarrow{P}_L||{}^{B}\overrightarrow{P}_R|}) \tag{5}$$

The angle β, using the projection of the optical axis $\overrightarrow{z}$ onto that plane, is computed from I_A by noting that the vector $\overrightarrow{C_AR_A}$ in F_A is necessarily perpendicular to this projection. From the above follows:

$$\beta = \frac{\pi}{2} - acos(\frac{{}^{A}\overrightarrow{P}_B \cdot \overrightarrow{C_AR_A}}{|{}^{A}\overrightarrow{P}_B||\overrightarrow{C_AR_A}|}), \tag{6}$$

with

$$ {}^{A}\overrightarrow{P}_{B} = [{}^{A}\mathbf{p}_{Bu} \quad {}^{A}\mathbf{p}_{Bv} \quad f_A] \tag{7}$$

where f_A is the focal length of $Camera_A$. In brief, the angle α is computed from the image I_B, and the angle β from the image I_A. Together, with the inter-marker distance d, we can compute the distance l between the cameras using eq. 8 as established in [12].

$$ l = \frac{d}{2\sin\alpha}\left(\cos\alpha\cos\beta + \sqrt{1-\cos^2\alpha\sin^2\beta}\right), \tag{8}$$

3.2 Extension to 3D problem

The solution to the full 3D problem consists in finding the rotation matrix **R** and translation matrix **T** between the frames of reference F_A and F_B. The following three constraints are used to estimate **R** and **T**:

- C_1: C_B is located with respect to F_A along the vector ${}^{A}\overrightarrow{P}_{B}$, at a distance l computed in Section 3.1, see eq. 8;

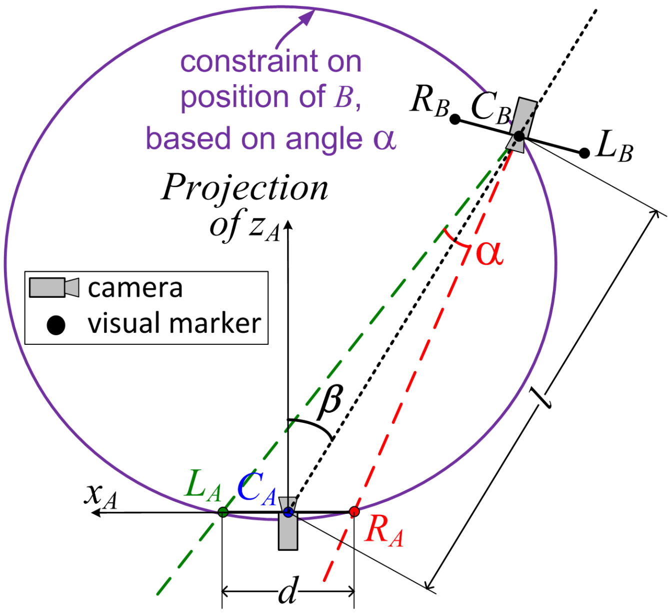

Fig. 2 Our 3D solution relies partially on a conversion into a 2D localization problem. The reduction in 2D is possible because C_A, L_A, R_A and C_B are all coplanar. Two angles are extracted within this plane: the angle α between the markers L_A and R_B as perceived by B, and the angle β, between C_B and the projection of the optical axis of A onto this plane.

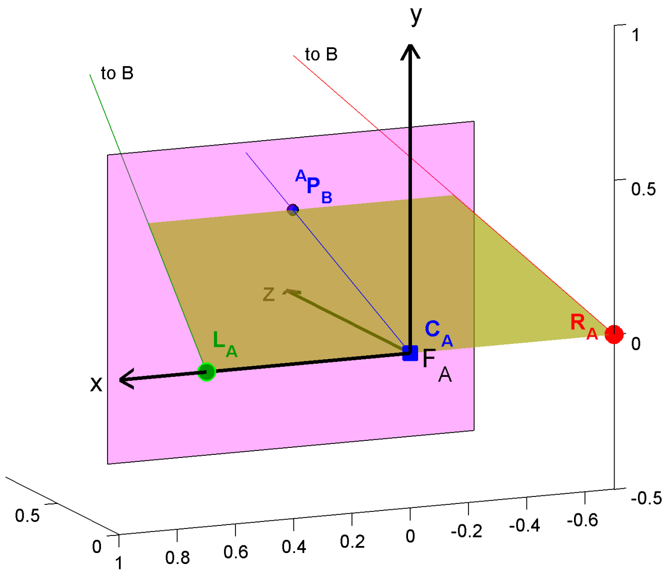

Fig. 3 The problem, as seen from the camera of $Robot_A$. The z-axis, which points away from reader, is the optical axis of the camera. Note that the x-axis must point left, to respect the right hand rule of Cartesian coordinate systems. The pink surface is the image plane and the point ${}^{A}\mathbf{p}_B$ lies within that image plane.

- C_2: the plane defined in F_B containing ${}^{B}\overrightarrow{P}_R$, ${}^{B}\overrightarrow{P}_L$ is spatially coplanar with the plane defined in F_A by ${}^{A}\overrightarrow{P}_B$ and $\overrightarrow{C_A R_A}$;
- C_3: the vector ${}^{A}\overrightarrow{P}_B$ defined in F_A and the vector ${}^{B}\overrightarrow{P}_A$ defined in F_B are spatially collinear and facing opposite directions.

Note that in both frames of references F_i, there is a plane passing through the landmarks of the opposite robots and also passing through the local coordinate (0,0,0).

We start by creating an intermediary frame of reference F'_B and making it coincide with F_A. F'_B is a copy containing all the points found in F_B. Our goal is to find the needed transformations so that, in the end, F'_B matches the location of F_B with respect to F_A.

Given that a plane can be uniquely defined by two linearly independent vectors, we have the plane in F_A (shown as yellow in Fig. 3) defined by the pair of vectors $\overrightarrow{C_A R_A}$ and ${}^{A}\overrightarrow{P}_B$, with the later the pixel location of B in the image plane of I_A. To satisfy constraint C_2, we must align these planes, via a rotation matrix $\mathbf{R_1}$. No translation is needed to align them, as they both pass through their respective frames' origins. We compute the required rotation matrix by finding the axis-angle rotation needed to align the normal vector $\overrightarrow{n}_B$ of the plane in F'_B with the normal vector $\overrightarrow{n}_A$ of the plane in F_A. The normal $\overrightarrow{n}_A$ is computed by the cross product:

$$\overrightarrow{n}_A = \overrightarrow{C_A R_A} \times {}^{A}\overrightarrow{P}_B. \tag{9}$$

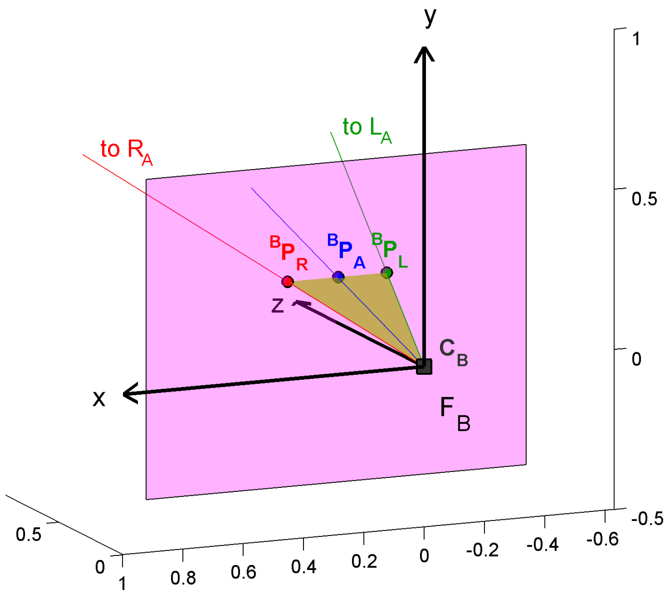

Fig. 4 The problem, as seen from the other robot *Robot*$_B$. The z-axis, which points away from reader, is the optical axis of camera B. This camera sees the two markers R_A and L_A, as well as the other camera A. The points ${}^B\mathbf{p}_R$, ${}^B\mathbf{p}_A$ and ${}^B\mathbf{p}_L$ all lie within the image plane depicted in pink.

The product will point upward, i.e. have a positive value for y, since the focal distance is always positive, i.e. $f_A > 0$. In F_B' we use the pair of vectors ${}^B\mathbf{P}_L$ and ${}^B\mathbf{P}_R$ to define the same physical plane (shown as yellow in Fig. 4). Its normal will be:

$$\overrightarrow{n}_B = {}^B\overrightarrow{P}_L \times {}^B\overrightarrow{P}_R. \tag{10}$$

The rotation matrix $\mathbf{R_1}$ is computed from Rodrigues' rotation formula:

$$\mathbf{R_1} = \text{Rodrigues}(\hat{e}_1, \Theta_1) \tag{11}$$

which is equivalent to

$$\mathbf{R_1} = \mathbf{I} + [\hat{e}_1]_x \sin\Theta_1 + (1 - \cos\Theta_1)(\hat{e}_1\hat{e}_1^T - \mathbf{I}) \tag{12}$$

with $\mathbf{I}$ being the $3x3$ identity matrix. The axis of rotation $\hat{e}_1$ is

$$\hat{e}_1 = \overrightarrow{n}_B \times \overrightarrow{n}_A \tag{13}$$

and the angle Θ_1 is

$$\Theta_1 = \text{acos}\Big(\frac{\overrightarrow{n}_B \cdot \overrightarrow{n}_A}{|\overrightarrow{n}_B||\overrightarrow{n}_A|}\Big). \tag{14}$$

With F_A and $\mathbf{R_1}F_B'$ having their planes coplanar, the next step is to align the vector $\mathbf{R_1}{}^B\overrightarrow{P}_A$ with the opposite vector $-{}^A\overrightarrow{P}_B$ with a rotation matrix $\mathbf{R_2}$, in order to satisfy constraint C_3. This rotation happens in the same plane defined earlier, so the rotation axis is $\hat{e}_2 = \overrightarrow{n}_A$. The angle of rotation is calculated as follows:

$$\Theta_2 = \mathrm{acos}\left(\frac{-{}^A\overrightarrow{P}_B \cdot \mathbf{R_1}{}^B\overrightarrow{P}_A}{|{}^A\overrightarrow{P}_B||{}^B\overrightarrow{P}_A|}\right) = \pi - \mathrm{acos}\left(\frac{{}^A\overrightarrow{P}_B \cdot \mathbf{R_1}{}^B\overrightarrow{P}_A}{|{}^A\overrightarrow{P}_B||{}^B\overrightarrow{P}_A|}\right). \tag{15}$$

The second rotation matrix will be:

$$\mathbf{R_2} = \mathrm{Rodrigues}(\hat{e}_2, \Theta_2). \tag{16}$$

Finally, we translate $\mathbf{R_2}\mathbf{R_1}F_B'$ by the distance l, along the ${}^A\overrightarrow{P}_B$ axis to meet the constraint C_1:

$$\overrightarrow{T} = l\frac{{}^A\overrightarrow{P}_B}{|{}^A\overrightarrow{P}_B|}, \tag{17}$$

with $\overrightarrow{T}$ being trivially convertible to the translation matrix, $\mathbf{T}$. We thus have found the transformation between frames F_A and F_B, and hence the relative pose estimate, since

$${}^AF_B = \mathbf{T}\mathbf{R_2}\mathbf{R_1}F_B', \tag{18}$$

with the position and orientation of F_B expressed in F_A's coordinate is ${}^AF_B$, and the transformed F_B' coincide with ${}^AF_B$ at this point.

To verify the correctness of the above formulae, we implemented a simulation in MATLAB. The algorithm was able to recover the position and orientation of $Robot_B$ relative to $Robot_A$. We also investigated the estimation of ${}^A\mathbf{p}_B$ and ${}^B\mathbf{p}_A$ in equations 1 and 2, which impacts the evaluation of β. The error on β was less than a tenth of a percent of the real β value at a distance of about 10 m, which resulted in an additional mean position and orientation errors of 0.57 *cm* and 0.05 deg, respectively. A discussion on noise models and their impact on real world applications is available in [12], which still apply to our 3D extension of the analytical solution via the distance l.

4 Experimental Setup

Two identical assemblies were constructed, each one comprising of a consumer-grade Logitech C905 camera and two white LEDs [1] as markers. The camera and the markers were securely fixed on an aluminum rod. The camera had a 75^o diagonal field of view and the LEDs had a viewing angle of 140^o. The distance between the markers were $d_A = 83.1$ *cm* and $d_B = 75.7$ *cm*. The optical axis of the camera was perpendicular to the rod. The rig A was placed on a table and used as the fixed frame of reference F_A.

[1] LED model SMP6-UWDW (4000 micro-candelas) from Bivar Inc

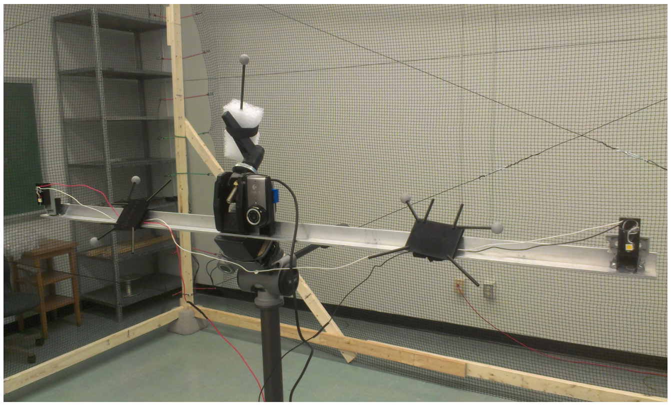

Fig. 5 Each robot was a long metal rod on which we attached a camera in the center and a LED at each end of the rod. We added multiple OptiTrack markers so the robot could be tracked by the OptiTrack V100:R2 cameras fixed all around the room. The moving robot is on a tripod with a swivel head, and a clamp is firmly holding the metal rod in place.

The C905 cameras use a rolling shutter acquisition technique, which introduces important distortion in the presence of camera motion. The second rig B was thus mounted on a standard camera tripod, in order to avoid the above mentioned distortions. Using this technique, the rig B was moved in 3D space and was oriented in arbitrary directions. The cameras were calibrated beforehand using the MATLAB Calibration Toolbox [2]. In order to ease the recognition of the LEDs on every picture, we adjusted the cameras' gain and exposure time so that only the LED markers were visible. The position of the markers R_i and L_i were found by extracting the centers of mass of the light spots in every image, and these locations became the input to the proposed 3D CL algorithm.

We used a separate OptiTrack system [3] to acquire ground truth. Four cameras were placed at the corners of the ceiling and a fifth one pointing toward the center of the room. To ensure optimal precision of the ground truth, we calibrated the system prior to the experiment using the OptiTrack Tracking Tools, with a reported calibration error of 0.13 *mm/marker*. OptiTrack markers were placed on the rig B for real-time tracking of the position and orientation. Note that these markers were not visible to $Camera_A$. Fig. 5 shows this setup. Because the reference from the OptiTrack system and F_A were different, we used a number of points in our dataset to find the transformation between these two references, via the Iterative Closest Point algorithm. This way, the ground truth data was referenced in F_A.

Seventy five distinct poses were tested by changing the position, orientation and/or height of the F_B between each test. The tripod could be placed between 3.5 m to 6.5 m away from the reference rig in order to stay in the zone covered by the OptiTrack setup. The height of the moving camera could be set between 1 m and 2 m

[2] http://www.vision.caltech.edu/bouguetj/calib_doc/index.html

[3] Model V100:R2 with five cameras; http://www.naturalpoint.com/optitrack/

off the ground. Lateral movements could not be over 3 *m* away from the center of the images taken by the reference camera, or else we would have placed $Camera_B$ outside the OptiTrack system's covering area.

5 Experimental Results

The first dataset was used for the computation of the matrices **T** and **R** needed to align the OptiTrack system's frame of reference with F_A. In this dataset, the sensor B was moved in all the available space and oriented in multiple directions, while ensuring that the two sensors were mutually observable. Fig. 6 shows a 2D perspective in the X-Z plane of the measured and estimated positions of the moving robot relatively to the static robot's reference coordinate system. Fig. 7a shows the error on the position estimation. The mean of this error is slightly below 2.5 *cm*. Fig. 7b presents the error on the orientation estimation. The mean of this error is 0.0370 *rad*, or about 2.12 *degrees*.

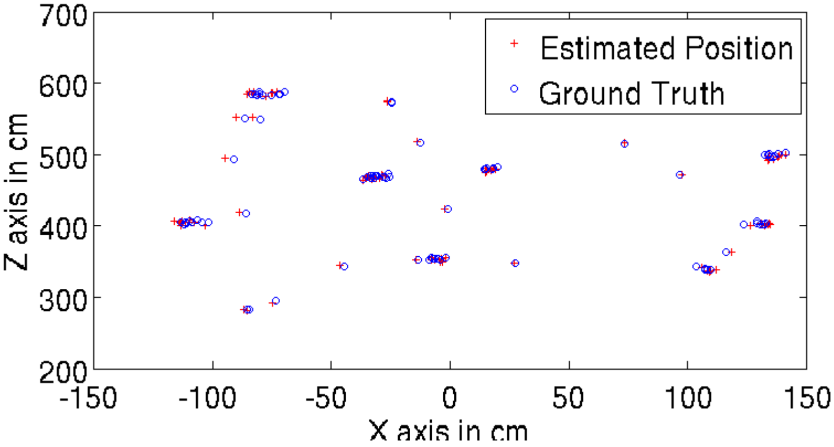

Fig. 6 A comparison of the position in a 2D X-Z perspective. The distance between the rigs vary between 3.5 m to 6.5 m due to the OptiTrack system limitations

For the second data set, Fig. 8a shows its position error, with a mean error of 1.9885 cm. Figure 8b shows its orientation error, with a mean error of 0.0205 rad, or 1.175 degrees. Figure 9 shows the 2D perspective in the X-Z plane of the measured and estimated positions of the moving camera setup.

Both test sets indicated that the orientation error is independent of the real orientation of the moving $Camera_B$.

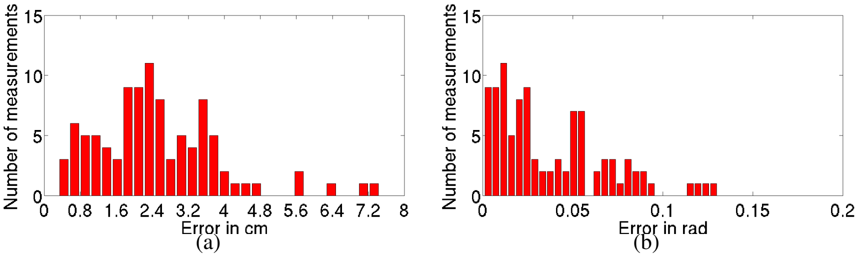

Fig. 7 (a) Histogram representing the position error made by the bearing-only system for the calibration dataset. The error mean is 2.4947 *cm*. (b) Histogram representing the orientation error made by the bearing-only system for the calibration dataset. The error mean is 0.0370 *rad*.

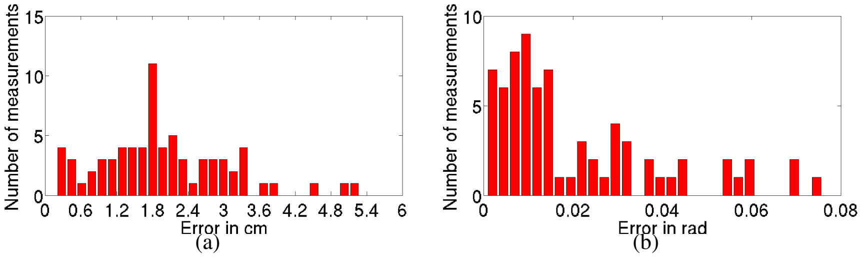

Fig. 8 (a) Histogram representing the position error made by the bearing-only system for the test dataset. The error mean is 1.9885 *cm*. (b) Histogram representing the orientation error made by the bearing-only system for the test dataset. The error mean is 0.0205 *rad*.

6 Non-Colinearity Analysis and Error Correction

In the previous sections, we presented an analytical solution to the 3D cooperative localization problem given that each robot's camera C_i was collinear to its markers R_i and L_i. In order to measure the impact of non-colinearity on the localization error, we implemented a simulation in MATLAB where we added small offsets to the position of the cameras. An offset applied to a camera is described by a translation of up to 35% of the distance d between a robot's markers. This translation is oriented in a direction ranging from 0 to π *rad* in the plane defined by the vectors ${}^{A}\overrightarrow{P}_{B}$ and $\overrightarrow{n}_{A}$ (equations 7 and 9, respectively). Thus, most realistic scenarios were coverred. In order to find a way to compute a correction to $Robot_B$'s pose estimation, we initially applied the offsets on only one camera. Table 6 above shows a summary of the impact of offset.

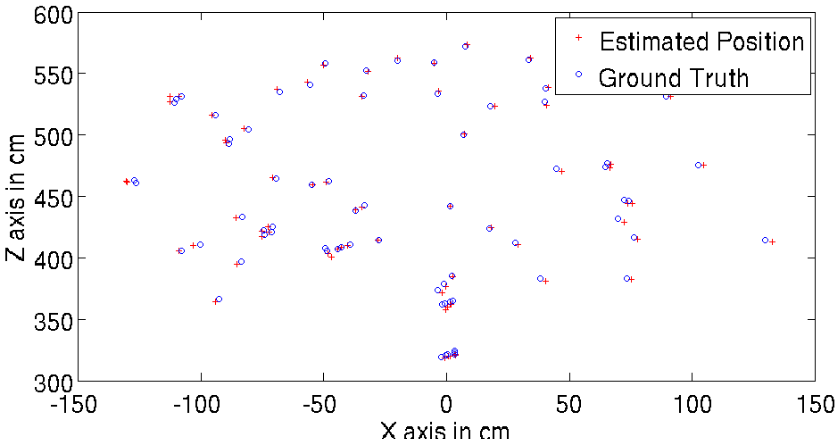

Fig. 9 A comparison of the position in a 2D X-Z perspective. The distance between the rigs vary between 3.5 *m* to 6.5 *m* due to the OptiTrack system limitations

Offset on	C_A Camera	C_B Camera
Impacts on	α	β
Impacts on	distance l	position
Correct **T** by	translation	rotation
Correct **R** by	small rotation	small rotation

When established that an offset on the camera of $Robot_A$ affects the estimation of the angle α. In turns, this impacts the estimation of the distance l between the two cameras (as in Eq. 8). In this case, the error can be reduced by a translation along vector $\overrightarrow{T}$ (see eq. 17) and a small orientation correction. To correct the estimated position, one needs to recover the angle Θ_A between the offset $\overrightarrow{S}_A$ of the camera C_A, and the vector ${}^A\overrightarrow{P}_B$, here estimated by $\overrightarrow{T}$:

$$\Theta_A = \mathrm{acos}\left(\frac{\overrightarrow{T}\cdot\overrightarrow{S}_A}{|\overrightarrow{T}||\overrightarrow{S}_A|}\right). \tag{19}$$

Using the estimated Θ_A, the estimated translation $\overrightarrow{T}$ can be corrected by a translation $\overrightarrow{T}_\varepsilon$, which is computed by finding the error on the distance, ε_l:

$$\varepsilon_l = |\overrightarrow{S}_A|\cos(\Theta_A) \tag{20}$$

$$\overrightarrow{T}_\varepsilon = \varepsilon_l \frac{\overrightarrow{T}}{|\overrightarrow{T}|}. \tag{21}$$

This is easily converted to a translation matrix, $\mathbf{T}_\varepsilon$. Similarly, Θ_A is used to compute a correcting rotation, $\mathbf{R}_\varepsilon^A$:

$$\mathbf{R}_{\varepsilon}^{A} = \text{Rodrigues}\left(\overrightarrow{L_A R_A}, \arcsin\left(\sin(\Theta_A)\frac{|\overrightarrow{S}_A|}{|\overrightarrow{T}| + \varepsilon_l}\right)\right). \tag{22}$$

On the other hand, when the camera on *Robot*$_B$ has an offset the estimation of the angle β is affected. This impacts the estimation of the position, but conserves a valid distance l estimation. In this case, the error can be reduced by a rotation about the R_A-L_A axis, again with a small orientation correction. To make these corrections, one needs to recover the angle Θ_B between the offset $\overrightarrow{S}_B$ of the camera C_B expressed in F_B, and the vector ${}^{B}\overrightarrow{P}_A$, here estimated by $-\overrightarrow{T}$, again expressed in F_B:

$$\Theta_B = \arccos\left(\frac{\mathbf{R_1}^{-1}\mathbf{R_2}^{-1}(-\overrightarrow{T})}{|\overrightarrow{T}|} \cdot \frac{\overrightarrow{S}_B}{|\overrightarrow{S}_B|}\right). \tag{23}$$

Using the estimated Θ_B, the estimated translation $\overrightarrow{T}$ and the estimated orientation $\mathbf{R_2 R_1}$ can be corrected by a rotation matrix $\mathbf{R}_{\varepsilon}^{B}$, which is computed by finding the error on the orientation, ε_θ:

$$\varepsilon_\theta = \arcsin(\sin(\Theta_B)\frac{|\overrightarrow{S}_B|}{|\overrightarrow{T}| + \varepsilon_l}) \tag{24}$$

$$\mathbf{R}_{\varepsilon}^{B} = \text{Rodrigues}(\overrightarrow{L_A R_A}, \varepsilon_\theta). \tag{25}$$

Since each camera offset has effects that are not related, we can sum the corrections independently, which leads to the revised equation 18:

$${}^{A}F_B = \mathbf{R}_{\varepsilon}^{B}\mathbf{T}_{\varepsilon}\mathbf{T}\mathbf{R}_{\varepsilon}^{B}\mathbf{R}_{\varepsilon}^{A}\mathbf{R_2}\mathbf{R_1}F_B'. \tag{26}$$

We tested Eq. 26 in a simulation at a distance $l = 11$ m. We applied offsets of 28 cm, or 35%d, in the direction of $\overrightarrow{n}_A$. This resulted in the greatest error during the estimation. After the correction, this simulation demonstrated that when the angle β is sharp, the position error could remain as high as 20 cm. This high error corresponds to the behavior that was observed in the 2D solution when the robots are observing each other at a sharp angle. However, when both cameras were not collinear and the *Robot*$_B$ was positionned at less than 45 deg away from the central point of *Robot*$_A$'s camera, one could expect a position error of less than 3cm and an orientation error of about 0.006 rad.

We compared the position errors due to our correction in Eq. 18 to the position errors induced by mislocating the markers in the image, since the latter will affect the estimation of α and β. For this comparison, we chose an angular noise of $\sigma_\varphi = 0.0003$ for the computation of β, as well as an angular noise of $\sigma_\varphi * \sqrt{2}$ for the measurement α. These values corresponded to an error of the estimation points in the image plane of approximately 0.3 $pixel$, close to what we observed in our system. The simulation results indicated that the average positional error for correction was generally less than 40% of the error due to the mislocation of the various landmarks. Consequently, our equation does not significantly affect the performance of the system. However, this error correction is biased, and it should be kept in mind if a filtering technique, such as EKF or UKF, is employed.

7 Conclusions and Future Work

In this paper we presented a novel analytical solution to the 3D Cooperative Localization problem using bearing only measurements between two robots. We derived the mathematical solution to the problem in three dimensions, which was verified with extensive simulations and experiments with real hardware. During experiments with a rig that moved freely in 6 DoF, our system demonstrated good position estimation (average error around 2 cm over 250 samples), despite the use of off-the-shelf consumer cameras and markers. This makes our solution particularly well suited for deployment on fleets of inexpensive robots.

The biggest challenge with the current physical implementation is to establish mutual observations, without interfering with the normal operation of the vehicles. Since the method is completely compatible with any type of cameras, omnidirectional cameras can be employed to alleviate this problem.

We are currently planning to apply this methodology to under-actuated square blimps that move at slow speeds [24], since the weight of the required hardware for our solution is very low. Applications to underwater vehicles [8] are also considered, since they are generally deployed in unstructured, GPS-denied environments. Another direction of research will be to employ Iterated Sigma Point Kalman Filtering [25] in order to integrate additional sensors and improve the accuracy of the state estimation.

References

1. Bahr, A., Leonard, J.J., Fallon, M.F.: Cooperative localization for autonomous underwater vehicles. The Int. Journal of Robotics Research **28**(6), 714–728 (2009)
2. Burgard, W., Moors, M., Fox, D., Simmons, R., Thrun, S.: Collaborative multi-robot exploration. In: Proc. of IEEE Int. Conf. on Robotics and Automation, vol. 1, pp. 476–481 (2000)
3. Cognetti, M., Stegagno, P., Franchi, A., Oriolo, G., Bülthoff, H.H.: 3-d mutual localization with anonymous bearing measurements. In: Proc. IEEE Int. Conf. on Robotics and Automation, pp. 791–798 (2012)
4. Cristofaro, A., Martinelli, A.: 3D cooperative localization and mapping: Observability analysis. In: American Control Conf. (ACC), 2011, pp. 1630–1635. IEEE (2011)
5. Davison, A.J., Kita, N.: Active visual localisation for cooperating inspection robots. In: Proc. of IEEE/RSJ Int. Conf. on Intel. Robots and Systems, vol. 3, pp. 1709–1715 (2000)
6. Dhiman, V., Ryde, J., Corso, J.J.: Mutual localization: Two camera relative 6-dof pose estimation from reciprocal fiducial observation. arXiv preprint arXiv:1301.3758 (2013)
7. Dieudonné, Y., Labbani-Igbida, O., Petit, F.: Deterministic robot-network localization is hard. IEEE Transactions on Robotics **26**(2), 331–339 (2010)
8. Dudek, G., Jenkin, M., Prahacs, C., Hogue, A., Sattar, J., Giguere, P., German, A., Liu, H., Saunderson, S., Ripsman, A., Simhon, S., L-A Torres, L.A., Milios, E., Zhang, P., Rekleitis, I.: A visually guided swimming robot. In: IEEE/RSJ Int. Conf. on Intel. Robots and Systems, pp. 3604–3609 (2005)

9. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24**(6), 381–395 (1981)
10. Franchi, A., Stegagno, P., Oriolo, G.: Probabilistic mutual localization in multi-agent systems from anonymous position measures. In: 49th IEEE Conf. on Decision and Control, p. 65346540. Atlanta, GA (2010)
11. Gage, D.W.: Minimum-resource distributed navigation and mapping. Tech. rep., DTIC (2000)
12. Giguere, P., Rekleitis, I., Latulippe, M.: I see you, you see me: Cooperative Localization through Bearing-Only Mutually Observing Robots. In: IEEE/RSJ Int. Conf. on Intel. Robots and Systems,, pp. 863–869. Portugal (2012)
13. Huang, G., Trawny, N., Mourikis, A., Roumeliotis, S.: Observability-based consistent ekf estimators for multi-robot cooperative localization. Autonomous Robots **30**(1), 99–122 (2011)
14. Kato, K., Ishiguro, H., Barth, M.: Identifying and localizing robots in a multi-robot system environment. In: Proc. of IEEE/RSJ Int. Conf. on Intel. Robots and Systems, vol. 2, pp. 966–971 (1999)
15. Kennedy, R., Daniilidis, K., Naroditsky, O., Taylor, C.: Identifying maximal rigid components in bearing-based localization. In: IEEE/RSJ Int. Conf. on Intel. Robots and Systems, pp. 194–201 (2012). DOI 10.1109/IROS.2012.6386132
16. Kukelova, Z., Bujnak, M., Pajdla, T.: Closed-form solutions to minimal absolute pose problems with known vertical direction. Computer vision–ACCV 2010 pp. 216–229 (2011)
17. Kurazume, R., Hirose, S.: An experimental study of a cooperative positioning system. Autonomous Robots **8**, 43–52 (2000)
18. Kurazume, R., Nagata, S., Hirose, S.: Cooperative positioning with multiple robots. In: IEEE Int. Conf. on Robotics and Automation, vol. 2, pp. 1250–1257 (1994)
19. Naroditsky, O., Daniilidis, K.: Optimizing polynomial solvers for minimal geometry problems. In: IEEE Int. Conf. on Computer Vision, pp. 975–982 (2011)
20. Rekleitis, I., Dudek, G., Milios, E.: Experiments in free-space triangulation using cooperative localization. In: IEEE/RSJ/GI Int. Conf. on Intel. Robots and Systems, pp. 1777–1782 (2003)
21. Rekleitis, I., Dudek, G., Milios, E.: Probabilistic cooperative localization and mapping in practice. In: IEEE Int. Conf. on Robotics and Automation, pp. 1907–1912 (2003)
22. Rekleitis, I.M., Dudek, G., Milios, E.E.: On multiagent exploration. In: Proc. of Vision Interface 1998, pp. 455–461. Vancouver, Canada (1998)
23. Roumeliotis, S., Bekey, G.: Collective localization: A distributed kalman filter approach to localization of groups of mobile robots. In: IEEE Int. Conf. on Robotics and Automation, pp. 2958–2965 (2000)
24. Sharf, I., Persson, M.S., St-Onge, D., Reeves, N.: Development of aerobots for satellite emulation, architecture and art. In: Proc. the 13th Int. Symp. on Experimental Robotics (2012)
25. Sibley, G., Sukhatme, G., Matthies, L.: The iterated sigma point kalman filter with applications to long range stereo. In: Proc. of Robotics: Science and Systems, pp. 263–270 (2006)
26. Spletzer, J.R., Taylor, C.J.: A bounded uncertainty approach to multi-robot localization. In: Proc. of IEEE/RSJ Int. Conf. on Intel. Robots and Systems, vol. 2, pp. 1258–1265 (2003)
27. Weber, B., Zeller, P., Kuhnlenz, K.: Multi-camera based real-time configuration estimation of continuum robots. In: IEEE/RSJ Int. Conf. on Intel. Robots and Systems, pp. 3350–3355 (2012)
28. Zhou, X., Roumeliotis, S.: Determining the robot-to-robot 3D relative pose using combinations of range and bearing measurements: 14 minimal problems and analytical solutions to 3 of them. In: Proc. IEEE/RSJ Int. Conf. on Intel. Robots and Systems, pp. 2983–2990 (2010)
29. Zhou, X., Roumeliotis, S.: Determining the robot-to-robot 3D relative pose using combinations of range and bearing measurements (part ii),. In: Proc. IEEE Int. Conf. on Robotics and Automation, pp. 4736–4743. Shanghai, China (2011)
30. Zhou, X., Roumeliotis, S.: Determining 3D relative transformations for any combination of range and bearing measurements. IEEE Transactions on Robotics (2012)