Abstract

This paper presents the complete calibration procedure of a multi-camera system for mobile robot motion registration. Optimization-based, purely visual methods for the estimation of the relative poses of the motion registration system cameras, as well as the relative poses of the cameras and markers placed on the mobile robot were proposed. The introduced methods were applied to the calibration of the system and the quality of the obtained results was evaluated. The obtained results compare favourably with the state of the art solutions, allowing the use of the considered motion registration system for the accurate reconstruction of the mobile robot trajectory and to register new datasets suitable for the benchmarking of indoor, visual-based navigation algorithms.

Keywords Visual Navigation, Calibration, Motion Capture, Benchmarking, Multi-camera System

1. Introduction

Multi-camera systems used for the tracking of markers, both natural and artificial, are commonly used in the field of computer vision. Such systems allow for the estimation of the pose of the marker attached to a moving object and for the reconstruction of its path from the series of such measurements. The trajectory that is carefully reconstructed using such data is often used for tracking and localization in the context of surveillance [1][2], wildlife monitoring [3] and finally, the tracking and localization of mobile robots [4][5]. One of the most important uses for multi-camera video data in the field of mobile robotics is the evaluation of mobile robot navigation algorithms. While benchmarking navigation algorithms in outdoor applications is in most cases performed using global positioning system (GPS) data [6][7][8], GPS localization accuracy degrades quickly as the signal quality gets worse. Furthermore, GPS positioning in general does not work inside buildings. In such a case, vision-based assessment of the accuracy of navigation algorithms becomes the most attractive solution. As such systems are often composed of multiple cameras, the calibration of such multi-camera setups and the accuracy of the reconstruction of the robot pose are crucial. Moreover, mobile robots are often equipped with multiple sensors and markers of different types. This is why the estimation of spatial relationships between the sensors and the markers, as well as their relation to the robot reference frame must be performed [9][10][11]. Failure to do so leads to gross errors.
In [12] and [13], the authors present multi-camera systems capable of tracking multiple robots. The systems use no or minimum artificial markers for the identification of the tracked objects and the tracking process allows only for the reconstruction of the paths of the robots with limited information on orientation. The proposed applications of the systems are constrained to the tracking of movement over a two-dimensional surface, and the goal of the calibration is to provide correct camera handoff. A markerless solution for 3D robot tracking is presented in [4]. The position of the robot in multiple views is detected by motion segmentation and the 3D localization is based on the minimization of a cost function. Camera calibration, both intrinsic and extrinsic, is necessary to properly place the tracked object in the 3D space. While the method is general, the accuracy is not sufficient for the ground-truth data generation. A different approach is presented in [14] and [15]. These methods rely on natural image features found in the registration environment. The cameras are calibrated using multiple view geometry methods followed by bundle adjustment. The solution presented in [14] is even capable of automatic discovery of the spatial relations between multiple cameras attached to a common, rigid frame. However, in order to do so it relies on odometry measurements. Marker-based multiple camera calibration is more commonly applied in multi-camera surveillance systems and is required for multi-camera object tracking and handoff [1][16]. However, in general, object tracking in such systems does not include orientation tracking, as such information is not necessary from the point of view of the end application.

Failure to provide the correct calibration data may cause large parts of the dataset to become unreliable and unusable. The dataset gathered during the Rawseeds Project [5] is one of the most widely used when it comes to the benchmarking of the simultaneous localization and mapping systems as it incorporates data from numerous sensors including four cameras. The procedure of the cameras’ relative poses estimation was only vaguely described in the project reports.

Unfortunately, the provided relative poses of the cameras significantly differ from the poses estimated from the video sequences, thus rendering the usefulness of the dataset doubtful. To confirm this, we performed a simple experiment using the images provided in the Rawseeds database. In this experiment, the relative pose of the cameras was estimated by matching image features detected and described using the ORB algorithm [17]. The 8-point algorithm [18] combined with the LMedS framework [19] was used to calculate the essential matrix, from which the rotation and translation components were extracted [20][18]. Figure 1 presents the feature matches used for the estimation of the essential matrix. Table 1 presents the rotation and the unitary translation vectors describing the transformation between two of the cameras used in the Rawseeds Project. There is a clear difference in the results given in the database description and the results returned by the algorithm estimating the scene structure. The same situation was observed for multiple pairs of images in the database.

<table>
<thead>
<tr>
<th>Camera</th>
<th>$t_x$</th>
<th>$t_y$</th>
<th>$t_z$</th>
<th>$r_x$</th>
<th>$r_y$</th>
<th>$r_z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>provided</td>
<td>0.0018</td>
<td>-0.0080</td>
<td>0.0056</td>
<td>-0.9997</td>
<td>-0.0034</td>
<td>0.0244</td>
</tr>
<tr>
<td>estimated</td>
<td>0.0431</td>
<td>0.0062</td>
<td>-0.0009</td>
<td>0.1396</td>
<td>-0.1274</td>
<td>-0.9820</td>
</tr>
</tbody>
</table>

Table 1. The rotation ($r_x$, $r_y$, $r_z$) and translation ($t_x$, $t_y$, $t_z$) between the cameras SVS_R and SVS_L of the Rawseeds Project

Figure 1. The feature matches used for essential matrix estimation

Figure 2. The motion capture system camera layout

The results of this experiment show that a precise description of the calibration process is an important part of the procedure, as without it is impossible to reliably reproduce the achieved results and identify the sources of possible discrepancies.

In this article, we present a method for the calibration of the multi-camera registration system for visual navigation benchmarking. The calibration allows us to capture the spatial relationships between the cameras observing the robot, thereby allowing for accurate path reconstruction. Moreover, the method for the estimation of the spatial arrangement of sensors and markers with respect to the mobile robot frame is also presented. The acquired data allows reliable benchmarking of the algorithms performing the task of simultaneous localization and mapping (SLAM) [21][22] or visual odometry [23][24]. As shown in [25], the presence of additional structured markers placed throughout the environment improves the accuracy of navigation in single robot scenarios. We believe that the introduction of additional unique artificial markers associated with individual robots, as presented in this article, will allow us to improve the accuracy in scenarios in which external cameras are used or in the case of collaborative SLAM.

2. Registration system

2.1. Motion capture subsystem architecture

The considered motion capture system (MCS) consists of five high-resolution Basler acA1600 cameras equipped...
with low-distortion, aspherical 3.5mm lenses. The cameras are mounted under the ceiling of the laboratory, forming an X-like structure. The field of view (FOV) of the central camera partially overlaps with the FOVs of all the peripheral cameras. Moreover, the FOV of each of the peripheral cameras partially overlaps with FOV of the two neighbouring peripheral cameras. Figure 2 presents the alignment of the cameras while Figure 3 presents the concept of overlapping FOVs.

The MCS coordinate system is coincident with the coordinate system of the central camera. The rotation matrix and translation vector of the peripheral cameras with regard to the MCS frame are defined as \( R_R^C \) and \( t_R^C \), where \( C \) stands for the central camera and \( P_i \) represents the \( i \)-th peripheral camera.

2.2 Multi-camera robot

A WiFiBot Lab V3 robot is used as the mobile platform in the registration system. In order to provide a wide field of view and to gather data for stereo-vision algorithms, two Basler acA640 cameras with 3.5mm lenses are installed on the robot using adjustable camera mounts. A rigid frame with a chessboard marker is attached to the robot, allowing us to track the robot’s position. Two additional circular markers are installed on the sides of the robot to facilitate research on the vision-based cooperation algorithms. The robot is shown in Figure 4.

As the MCS provides the information on the position of the chessboard marker, it is assumed that the robot’s coordinate system is defined by the chessboard marker. The pose of the on-board cameras with regard to the robot’s frame is defined by the rotation matrix \( R_{i}^{R} \) and the translation vector \( t_{i}^{R} \), where \( R \) stands for the robot and \( i \) is the frame of the \( i \)-th robot camera. Similarly, the rotation matrix and translation vector of the circular markers are defined as \( R_{R}^{M_{i}} \) and \( t_{R}^{M_{i}} \), with \( M_{i} \) standing for the \( i \)-th marker.

2.3 Data registration

The registration system used for ground truth data gathering is set up to achieve maximum possible accuracy.

The snapshots taken every 100 ms by both the overhead and robot-mounted cameras are synchronized to under 1 ms. Thanks to such a solution, there is no need to interpolate the measurement data, which is an advantage over timestamp-based solutions presented in [5][7][26]. Synchronized acquisition of uncompressed images at such data rates is a challenging task. To provide bottleneck-free communication and registration, dedicated GigE Vision acquisition cards were used for the data transmission and SSD, and RAM-disks were used as storage. The detailed presentation of the whole system used for data acquisition is outside of the scope of this article and can be found in [27].

3. Camera calibration

The camera calibration is a necessary step in 3D computer vision allowing the extraction of metric information from 2D images. The intrinsic parameters and the distortion coefficients of the cameras used in the registration system are estimated independently for each camera using the method proposed in [28]. The input data for the method are the images of a planar pattern with distinct features observed by the camera shown at several different orientations. The orientation change can be performed by moving either the camera or the marker, yet no prior knowledge on the motion is necessary for correct operation. The calibration procedure assumes that the 3D world points are projected on the camera image plane according to the rational camera model described in [29].

The projection function is defined as:

\[
q = h(p, A, D) = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \begin{bmatrix} k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 \end{bmatrix} p_1 p_2
\]

where \( p = [x, y, z]^T \) stands for the Cartesian position of the point in the camera coordinates, \( q = [u, v]^T \) is its position on the image plane, \( A \) is the camera matrix, \( k_i \) is the \( i \)-th radial distortion coefficient and \( p_i \) stands for the
i-th tangential distortion coefficient. The elements of the matrix $A$ are the focal lengths in the $x$ and $y$ directions denoted as $f_x$, $f_y$ (as in the general case the pixels can be non-square), and the coordinates of the principal point of the camera are denoted as $c_x$ and $c_y$. The image position of a 3D point $p$ is calculated according to the following formulae:

$$
x' = \frac{x}{z} + c_x$$
$$
y' = \frac{y}{z} + c_y$$
$$
r^2 = x'^2 + y'^2$$
$$
R = \frac{1 + k_1 r^2 + k_2 r^4 + k_3 r^6}{1 + k_4 r^2 + k_5 r^4 + k_6 r^6}$$
$$
x'' = x' R + 2 p_1 x'y' + p_2 (r^2 + 2 x'^2)$$
$$
y'' = y' R + 2 p_1 (r^2 + 2 y'^2) + p_2 x'y'$$
$$
u = f_x x'' + c_x$$
$$
v = f_y y'' + c_y$$

The image coordinates of the chessboard marker’s corners detected on the images registered using the camera that is to be calibrated are used as the input data. Prior knowledge on the arrangement of the feature points on the planar pattern enables us to establish the correspondences between the observed features and the pattern features. The goal of the procedure is to find the set of intrinsic parameters and the distortion coefficients that assure the coincidence of these two sets of features. The procedure starts with finding a closed-form solution, which is then used as the starting point for nonlinear refinement based on the maximum likelihood criterion. The implementation of the algorithm is readily available in the OpenCV image processing library [30].

As required by the method, a planar marker with a chessboard pattern is used for the calibration. A number of calibration marker images are gathered for each of the cameras covering a diverse set of relative marker-camera poses. Half of the images are used as a calibration set, while the other half are used only for testing. Figure 5 presents the exemplary images used for the calibration.

The estimated intrinsic parameters and the average reprojection error of the MCS and the robot cameras, as well as the additional camera ($E$) used in the robot calibration (see Section 4.2) are presented in Table 2.

4. Methodology

4.1. Motion capture system calibration

The knowledge of the exact position of the cameras with regard to the MCS coordinates coincident with the central camera coordinates is required in order to precisely track the robot’s position. As the FOVs of all the peripheral cameras partially overlap with the FOV of the central camera, the simplest way to estimate their relative position is to use the available stereo-calibration procedures. Unfortunately, such an approach can lead to inconsistency of the relative position of the peripheral cameras, as the position of each camera is estimated independently.

In order to overcome this problem, a holistic approach to simultaneously estimate the pose of all the peripheral cameras was developed. The proposed method uses observations of a chessboard calibration marker visible by at least two cameras at the same moment (Figure 6).

The model parameters vector $\beta_{MCS}$ consists of the poses of the peripheral cameras and the poses of the calibration marker during consequent observations:

$$
\beta_{MCS} = \left[ r_{P_1} t_{P_1} \ldots r_{P_i} t_{P_i} r_{M_1} t_{M_1} \ldots r_{M_N} t_{M_N} \right]^{T} 
$$

where $r_{P_i}$ and $t_{P_i}$ stand for the orientation described using the Rodrigues’ rotation formula and translation vector of the i-th peripheral camera respectively. Analogously, $r_{M_i}$ and $t_{M_i}$ represent the pose of the calibration marker during the i-th observation. The Rodrigues’ rotation representation is selected as the elements of the rotation vector can have any numerical value, thus the optimization problem remains unconstrained as opposed to the unitary quaternion or rotation matrix representations.
### Table 2. The intrinsic parameters of the cameras used

<table>
<thead>
<tr>
<th>Camera</th>
<th>cal. error</th>
<th>test error</th>
<th>$f_x$</th>
<th>$f_y$</th>
<th>$c_x$</th>
<th>$c_y$</th>
<th>$k_1$</th>
<th>$k_2$</th>
<th>$k_3$</th>
<th>$k_4$</th>
<th>$k_5$</th>
<th>$k_6$</th>
<th>$p_1$</th>
<th>$p_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>0.308</td>
<td>0.311</td>
<td>827.08</td>
<td>826.69</td>
<td>794.83</td>
<td>590.31</td>
<td>1.2086</td>
<td>0.1021</td>
<td>0.0365</td>
<td>1.2976</td>
<td>-0.0333</td>
<td>0.0916</td>
<td>-0.0009</td>
<td>0.0003</td>
</tr>
<tr>
<td>P₁</td>
<td>0.354</td>
<td>0.363</td>
<td>827.45</td>
<td>827.23</td>
<td>796.83</td>
<td>593.34</td>
<td>1.0454</td>
<td>-0.0031</td>
<td>0.1194</td>
<td>0.0903</td>
<td>-0.0987</td>
<td>0.1636</td>
<td>0.0007</td>
<td>0.0001</td>
</tr>
<tr>
<td>P₂</td>
<td>0.321</td>
<td>0.312</td>
<td>823.08</td>
<td>822.98</td>
<td>799.61</td>
<td>617.20</td>
<td>0.0035</td>
<td>0.1056</td>
<td>0.0433</td>
<td>0.0462</td>
<td>0.0160</td>
<td>0.0835</td>
<td>0.0003</td>
<td>-0.0007</td>
</tr>
<tr>
<td>P₃</td>
<td>0.339</td>
<td>0.311</td>
<td>829.46</td>
<td>828.96</td>
<td>788.31</td>
<td>608.16</td>
<td>1.4200</td>
<td>0.0999</td>
<td>0.0198</td>
<td>1.4932</td>
<td>-0.0309</td>
<td>0.0739</td>
<td>0.0003</td>
<td>-0.0007</td>
</tr>
<tr>
<td>P₄</td>
<td>0.331</td>
<td>0.325</td>
<td>830.74</td>
<td>830.47</td>
<td>805.74</td>
<td>624.86</td>
<td>1.7596</td>
<td>0.0997</td>
<td>0.0324</td>
<td>1.8364</td>
<td>-0.0376</td>
<td>0.0864</td>
<td>-0.0001</td>
<td>0.0004</td>
</tr>
<tr>
<td>R₁</td>
<td>0.279</td>
<td>0.337</td>
<td>653.69</td>
<td>654.37</td>
<td>355.81</td>
<td>240.47</td>
<td>-0.0483</td>
<td>0.9346</td>
<td>0.4307</td>
<td>0.3516</td>
<td>0.7350</td>
<td>0.9659</td>
<td>-0.0005</td>
<td>-0.0007</td>
</tr>
<tr>
<td>R₂</td>
<td>0.218</td>
<td>0.176</td>
<td>655.14</td>
<td>655.60</td>
<td>334.18</td>
<td>264.61</td>
<td>0.1514</td>
<td>0.7149</td>
<td>0.7822</td>
<td>0.1760</td>
<td>-0.1514</td>
<td>0.0916</td>
<td>0.0003</td>
<td>-0.0006</td>
</tr>
<tr>
<td>E</td>
<td>0.412</td>
<td>0.473</td>
<td>1081.71</td>
<td>1080.86</td>
<td>641.99</td>
<td>373.12</td>
<td>0.0904</td>
<td>-0.7019</td>
<td>4.5867</td>
<td>-0.1190</td>
<td>4.3336</td>
<td>0.0005</td>
<td>-0.0005</td>
<td>0.270</td>
</tr>
</tbody>
</table>

The Levenberg-Marquardt algorithm [31][32] is used to minimize the root mean square (RMS) of distances between the observed and the predicted positions of chessboard marker corners on the registered images:

$$S_{MCS}(\beta_{MCS}) = \sum_{c,i,j} (o^j_c - h\left(cP^j_i, A_c, D_c\right))^2$$  \hspace{1cm} (13)

where $o^j_c$ is the position of the $j$-th calibration marker point on the $c$-th image registered by the camera $c \in \{C, P₁, P₂, P₃, P₄\}$, $cP^j_i$ is the position of the point in the camera coordinates, $A_c$ and $D_c$ are camera matrix and distortion coefficients of the camera $c$.

The position of the $j$-th calibration marker point in the cameras’ coordinate frames for $i$-th pose of the calibration marker is defined as:

$$cP^j_i = R(r_{M_i})p^j_i + t_{M_i}$$  \hspace{1cm} (14)

$$p^j_i = R(r_{R_i})T\left(R(r_{M_i})p^j_i + t_{M_i} - t_{R_i}\right)$$  \hspace{1cm} (15)

where $R(r)$ stands for the rotation matrix defined by the Rodrigues’ rotation vector $r$ and $p^j_i$ is the position of the $j$-th point in the marker coordinates.

It is worth noting that each of the predicted point projections depends only on the pose of the calibration marker during the particular observation and optionally on the pose of one of the peripheral cameras. Therefore, the Jacobian matrix of the minimized function over the parameters vector $\beta_{MCS}$ is sparse. Moreover, its elements can be calculated analytically which significantly speeds up the optimization procedure.

### 4.2. Robot camera calibration

The MCS allows us to precisely track the pose of the calibration marker attached to the mobile robot. In order to track the pose of the on-board cameras, it is necessary to calculate their position with regard to the robot’s calibration marker. As mechanical measurements would be cumbersome and imprecise, a visual method using an external camera and an additional calibration marker is proposed. The additional marker is placed in the field of view of the on-board cameras and the external camera is used to register an image with both the robot and the additional marker visible (Figure 7). A diverse set of images with different marker and external camera poses is registered (Figure 8).

The model parameters vector $\beta_{RC}$ consists of the poses of the robot cameras and the poses of both the external marker and the external camera during consecutive observations:

$$\beta_{RC} = [r_{R₁}, t_{R₁}, r_{R₃}, t_{R₃}, r_{M₁}, t_{M₁}, \ldots, r_{Mₙ}, t_{Mₙ}, r_{E₁}, t_{E₁}, \ldots, r_{Eₙ}, t_{Eₙ}]$$  \hspace{1cm} (16)
where \( r_R \) and \( t_R \) stand for the orientation described using the Rodrigues’ rotation formula and the translation vector of the \( i \)-th robot camera respectively. Analogously, \( r_M, t_M, r_E, t_E \) represent the pose of the external calibration marker and the external camera during the \( i \)-th observation.

Similarly to the MCS calibration, the Levenberg-Marquardt algorithm\[31\]\[32\] is used to minimize the RMS of distances between the observed and the predicted positions of marker corners on the registered images:

\[
S_{RC}(\beta_{RC}) = \sum_{c, i, j} \left( c q_j^i - h (c p_j^i, A_c, D_c) \right)^2
+ \sum_{i, j} \left( Q_j^i - h (p_j^i, A_E, D_E) \right)^2
\] (17)

where \( c q_j^i \) is the position of the \( j \)-th external calibration marker point on the \( i \)-th image registered by the camera \( c \in \{ R_1, R_2, E \} \), \( c p_j^i \) is the position of the point in the camera coordinates, \( Q_j^i \) stands for the position of the \( j \)-th point of the robot marker on the \( i \)-th image registered by the external camera, and \( P_j^i \) is the position of this point in the external camera coordinates.

The position of the \( j \)-th external marker point in the camera \( c \) coordinate systems is defined as:

\[
e p_j^i = R(r_c) \left( R(r_M) p_j^i + t_M - t_c \right)
\] (18)

and the position of the \( j \)-th robot marker points in the external camera is calculated according to:

\[
P_j^i = R(r_E) \left( p_j^i - t_E \right)
\] (19)

Similarly to the MCS calibration, each of the predicted projections of the external marker points depends only on the pose of the external marker during the particular observation and on the pose of one of the cameras. The projections of the robot marker points depend only on the pose of the external camera during the particular observation. Therefore, the Jacobian matrix of the minimized function over the parameters vector \( \beta_{RC} \) is sparse. Its elements can also be calculated analytically which significantly speeds up the optimization procedure.

\[
S_{RM}(\beta_{RM}) = \sum_{c, i, j} \left( c q_j^i - h (c p_j^i, A_c, D_c) \right)^2
+ \sum_{i, j} \left( Q_j^i - h (p_j^i, A_E, D_E) \right)^2
\] (21)

4.3. Robot marker calibration

In order to use the circular patterns in the cooperative, vision-based algorithms, their exact pose with regard to the robot’s coordinate frame has to be established. The relative pose is calculated using a series of images with both the chessboard marker and the circular marker visible. A set of images taken at different camera poses was recorded (Figure 9).

The parameters vector \( \beta_{RM} \) of the model consists of the pose of the circular pattern marker and the poses of camera during subsequent observations:

\[
\beta_{RM} = \left[ r_M \ t_M \ r_E \ t_E \ r_E \ t_E \cdots r_E \ t_E \right]
\] (20)

where \( r_M \) and \( t_M \) stand for the Rodrigues’ rotation and translation vectors of the \( k \)-th robot marker respectively, while \( r_E \), \( t_E \) represent the pose of the chessboard marker with regard to the external camera’s frame during the \( i \)-th observation.

The Levenberg-Marquardt algorithm\[31\]\[32\] is used to minimize the RMS of differences between the observed and the predicted positions of both the chessboard marker corners and the centres of the circular pattern on the registered images:
The position of the camera coordinate systems is defined as:

\[ \mathbf{q}_j = \begin{bmatrix} r_x & r_y & r_z \end{bmatrix}^T, \quad \mathbf{t}_j = \begin{bmatrix} t_x[m] & t_y[m] & t_z[m] \end{bmatrix}^T \]

where \( \mathbf{q}_j \) is the position of the \( j \)-th chessboard marker corner on the \( i \)-th image registered by the external camera \( \mathbf{p}_i^j \) is the position of the point in the camera coordinates, \( Q_i^j \) stands for the position of the \( j \)-th circle centre of the circular marker on the \( i \)-th image registered by the external camera, and \( \mathbf{P}_j \) is the position of this point in the external camera coordinates.

The position of the \( j \)-th chessboard corner in the external camera coordinate systems is defined as:

\[ \mathbf{p}_i^j = \mathbf{R}(\mathbf{r}_E^j)\mathbf{p}_i^j + \mathbf{t}_E^j \]  \hspace{1cm} (22)

and the position of the \( j \)-th circular marker centre in the external camera is calculated according to the following formula:

\[ \mathbf{p}_i^j = \mathbf{R}(\mathbf{r}_E^j) \left( \mathbf{R}(\mathbf{r}_M^j)\mathbf{p}_i^j + \mathbf{t}_M \right) + \mathbf{t}_E^j \]  \hspace{1cm} (23)

The position of the chessboard corners depends only on the relative pose of the camera, while the position of the circular pattern points depend on both the pose of the camera and the pose of the circular marker. The Jacobian with regard to the model parameters is sparse and easily computable which significantly facilitates the optimization process.

5. Experiments and results

5.1. Motion capture system calibration

The data required to perform the calibration of the MCS was gathered by registering the images of a 5 × 8 chessboard calibration marker in 362 different poses in which it was observed by at least two cameras. Half of those images were used to calibrate the MCS according to the procedure described in Section 4.1. The other half, constituting the first testing set, was used for the evaluation of the obtained results. Additionally, the images of the mobile robot carrying a 4 × 6 chessboard marker in 180 poses were registered forming the second testing set. The initial poses of the marker w.r.t. the cameras’ coordinate systems required for the calibration procedure were established independently for each image using the Levenberg-Marquardt algorithm[31][32].

Figure 10 presents the estimated poses of the MCS cameras as well as the poses of the marker used for calibration. The numerical values of the obtained rotation and translation vectors are given in Table 3.

The RMS of the chessboard pattern corners’ reprojection error on the calibration set equalled 0.1487 pixel. The two testing sets were used to evaluate the quality of calibration results. The mean discrepancy of the chessboard marker corners’ 3D position estimated from the images registered by different cameras and recalculated to the global coordinates was selected as a quality measure:

\[ c_{c_i} = \begin{bmatrix} R_{c} & 0 \end{bmatrix} \begin{bmatrix} R_{c}^M & 0 \\ 0 & I \end{bmatrix} + t_j \]  \hspace{1cm} (24)

\[ \text{disc}(c_1, c_2, i, j) = ||c_{c_{i_1}} - c_{c_{i_2}}|| \]  \hspace{1cm} (25)

where \( c_{c_i} \) is the position of the \( i \)-th calibration pattern point observed at \( i \)-th image by the camera \( c \), \( R_{c}^M \) and \( I \) describe the transformation between the camera \( c \) and the central camera estimated by using the proposed calibration method (for \( c = C R_{c}^M + I \) is an identity matrix and \( I \) is a zero vector). \( R_{c}^M \) and \( I \) describe the pose of the marker with regard to the frame estimated as a solution to the PnP problem from the \( i \)-th frame by using algorithm described in [33]. The \( c_1 \) and \( c_2 \) are two cameras observing the marker at the same time. Figure 11 presents the histogram of point position discrepancy obtained for the first testing set, while the Figure 12 presents the histogram of discrepancy for the marker placed on the robot.

The mean discrepancy between the robot marker corners’ positions estimated from images registered by different cameras equalled 0.013[m] (Table 4) which allows reliable tracking of the robot’s position and calculating its ground truth position.

5.2. Robot camera calibration

A set of images was recorded with the robot’s on-board cameras and the external camera in 50 combinations of the external marker and the external camera poses. 40 of the image groups were used for calibration while the rest was used to evaluate the results. The method presented in Section 4.2 is used to determine the pose of the robot cameras. The final RMS of the reprojection error equalled 0.02 pixel. Figure 13 presents the 3D plot of relative camera and marker poses, while Table 5 contains the numerical values on the robot’s cameras position and orientation.

In order to evaluate the calibration process, the mean discrepancy between the estimated 3D positions of the external chessboard pattern corners with regard to the robot’s coordinates was calculated.

Table 3. The rotation \( (r_x, r_y, r_z) \) and translation \( (t_x, t_y, t_z) \) vectors of the peripheral cameras obtained using the proposed method

<table>
<thead>
<tr>
<th>Camera</th>
<th>( r_x )</th>
<th>( r_y )</th>
<th>( r_z )</th>
<th>( t_x[m] )</th>
<th>( t_y[m] )</th>
<th>( t_z[m] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P_1 )</td>
<td>0.2324</td>
<td>-0.2251</td>
<td>3.1024</td>
<td>1.2664</td>
<td>1.7380</td>
<td>0.0565</td>
</tr>
<tr>
<td>( P_2 )</td>
<td>-0.1267</td>
<td>0.1700</td>
<td>-0.0565</td>
<td>-1.3681</td>
<td>-1.6254</td>
<td>0.0372</td>
</tr>
<tr>
<td>( P_3 )</td>
<td>-0.1979</td>
<td>0.0891</td>
<td>0.0288</td>
<td>2.2425</td>
<td>-1.7965</td>
<td>-0.0932</td>
</tr>
<tr>
<td>( P_4 )</td>
<td>0.1177</td>
<td>-0.3493</td>
<td>3.0357</td>
<td>2.3493</td>
<td>1.6278</td>
<td>-0.0726</td>
</tr>
</tbody>
</table>

Table 4. The mean, standard deviation and maximum discrepancy

<table>
<thead>
<tr>
<th>Camera</th>
<th>( r_x )</th>
<th>( r_y )</th>
<th>( r_z )</th>
<th>( t_x[m] )</th>
<th>( t_y[m] )</th>
<th>( t_z[m] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( K_1 )</td>
<td>-1.170</td>
<td>1.115</td>
<td>1.248</td>
<td>0.336</td>
<td>0.066</td>
<td>-0.157</td>
</tr>
<tr>
<td>( K_2 )</td>
<td>-1.142</td>
<td>-1.170</td>
<td>-1.183</td>
<td>-0.037</td>
<td>0.063</td>
<td>-0.160</td>
</tr>
</tbody>
</table>

Table 5. The rotation \( (r_x, r_y, r_z) \) and translation \( (t_x, t_y, t_z) \) vectors of the robot cameras obtained using the proposed method; [m] stands for metres.
Figure 10. The estimated poses of the cameras and calibration marker with regard to the central camera; [m] stands for metres

Figure 11. The histogram of points position discrepancy in the testing set

Figure 12. The histogram of points position discrepancy for the robot marker tracking
Figure 13. The result of robot cameras calibration: robot marker - black, robot cameras - blue, external camera poses - red, external marker poses - green; [m] stands for metres

Figure 14. The histogram of points position discrepancy between the robot cameras; [mm] stands for millimetres

\[
\begin{align*}
\varepsilon_{c_i} & = R_c R_i (R_c^B c_1^B + i t_c^B) + t_c^C \\
\varepsilon_{E_i} & = R_c R_i (R_c^B c_2^B + i t_c^B) + i t_c^E
\end{align*}
\]  

(26)  

(27)

disc(c_1, c_2, i, j) = || c_i P_{j} - c_2 P_{j} ||

(28)

where \( c_i P_{j} \) is the position of the \( j \)-th calibration pattern point observed at \( i \)-th image by the robot camera \( c \), \( R_c R_i \) and \( t_c^C \) describe the estimated transformation between the robot camera \( c \) and the robot coordinates. \( R_c^B c_1^B \) and \( t_c^M \) describe the pose of the marker with regard to the camera frame estimated as a solution to the PnP problem from a frame \( i \) using algorithm described in [33]. The \( c_1 \) and \( c_2 \) are two cameras observing the marker at the same time. Figure 14 shows the histogram of discrepancy between point positions estimated by the robot cameras. Figure 15 shows the histogram of discrepancy between the point positions estimated with the robot cameras and the external cameras.

5.3. Robot marker calibration

The pose of each of the circular patterns was calculated independently. For each pattern a set of 50 images...
with both the chessboard pattern and the circular pattern marker of interest was recorded. The procedure described in Section 4.2 was applied to determine the pose of the circular pattern markers. The final RMS of the reprojection error equalled 0.021 pixel for the right marker and 0.025 pixel for the left pattern. Figures 16 and 17 show the relative poses of the markers and the external camera, while Table 6 contains the values of the marker translation and rotation vectors.

<table>
<thead>
<tr>
<th>Camera</th>
<th>( r_x )</th>
<th>( r_y )</th>
<th>( r_z )</th>
<th>( t_x [m] )</th>
<th>( t_y [m] )</th>
<th>( t_z [m] )</th>
</tr>
</thead>
<tbody>
<tr>
<td>right</td>
<td>1.209</td>
<td>1.204</td>
<td>1.197</td>
<td>0.406</td>
<td>0.161</td>
<td>-0.076</td>
</tr>
<tr>
<td>left</td>
<td>1.226</td>
<td>-1.204</td>
<td>-1.222</td>
<td>-0.102</td>
<td>-0.021</td>
<td>-0.077</td>
</tr>
</tbody>
</table>

Table 6. The poses of the circular pattern markers obtained using the proposed method; [m] stands for metres

6. Conclusions

This paper presents the calibration procedures of the three main components of the mobile robot movement registration system:

- estimation of relative poses of motion capture system cameras,
- estimation of robot camera poses with regard to the marker attached to the robot,
- estimation of additional marker poses with regard to the robot coordinates.

All the proposed methods use visual observations of the calibration markers and employ a Levenberg-Marquardt optimization algorithm to minimize the RMS of the reprojection error. The use of an additional, robot-mounted marker allows us to recover both the position and orientation of the robot with regard to the global reference frame with very good accuracy.

When compared to markerless methods, e.g., [4], the accuracy is over an order of magnitude better. Moreover, unlike the solutions presented in [13] and [12], it facilitates full pose recovery. The reliable observation of the robot-mounted marker may however be impossible or unfeasible in vast spaces – in such a case the calibration and tracking solutions based on natural features like [14] or [15] are a better choice, provided that there are enough natural features available for tracking. It is also worth noting that the methods based on multiple view geometry and bundle adjustment require the cameras to be calibrated to move [14][15]. Additional data, such as wheel odometry measurements, may also be required for the discovery of the extrinsic parameters of the camera system [14]. On the contrary, the presented system is completely passive, as the observations are performed by environment-bound sensors, i.e., the ceiling cameras.

The obtained calibration precision will allow the gathering of a reliable dataset for evaluation of indoor mono- and stereo-based visual navigation algorithms. The data used for the calibration will be made freely available along with the gathered robot trajectories in order to facilitate result reproducibility and the development of different calibration methods.

7. Acknowledgement

This research was financed by the Polish National Science Centre grant funded according to decision DEC-2011/01/N/ST7/05940, which is gratefully acknowledged.

8. References


