Emerald Industrial Robot

An experimental study on feature-based slam for multi-legged robots with RGB-D sensors

Journal:	*Industrial Robot*
Manuscript ID	IR-11-2016-0340
Manuscript Type:	Original Manuscript
Keywords:	3D, Machine vision, SLAM, Walking robots, RGB-D sensors

An experimental study on feature-based slam for multi-legged robots with RGB-D sensors

1 Introduction

The advent of compact, lightweight, and inexpensive RGB-D sensors gave rise to new research efforts in 3-D visual Simultaneous Localization and Mapping (SLAM). Direct depth measurements, readily available from such sensors as PrimeSense Carmine, Asus Xtion, Intel RealSense, Microsoft Kinect or Kinect 2 allow solving many of the problems inherent to SLAM solutions based on monocular sensing (Mur-Artal and Tardos, 2016). In addition, RGB-D sensors are easier to use on mobile robots, particularly those of limited size and payload than passive stereo cameras that require accurate calibration and high computing power for real-time processing of stereo images (Belter *et al.*, 2016b). Besides, the second generation time-of-flight RGB-D sensor Microsoft Kinect 2 can be employed also for outdoor SLAM (Belter *et al.*, 2015b).

One prominent class of mobile robots that require accurate pose estimates in 3-D and with respect to six degrees of freedom are multi-legged machines. Recent advancements in motion planning and control algorithms for legged locomotion enable multi-legged walking robots to be considered for Urban Search And Rescue (USAR) missions. The localization module is a crucial part of an autonomous walking robot navigation software. It was already demonstrated that planning footsteps and leg movements taking into account the perceived terrain map brings better results than simple reflexes, which often lead to repetitive, the trial-and-error behavior of the robot (Belter *et al.*, 2016b). On the other hand, localization with the accuracy of about the dimensions of the robot's foot is required to ensure proper registration of the sensory data into a terrain map and foothold planning (Belter and Skrzypczynski, 2013). Thus, a reliable and´ accurate 3-D SLAM solution is of pivotal importance to multi-legged walking robots in realworld applications.

Therefore, we evaluate in this article four state-of-the-art open-source SLAM systems. We focus exclusively on feature-based SLAM approaches that utilize RGB-D input. Mur-Artal *et al.* already pointed out the benefits of the feature-based approach in the context of monocular SLAM (Mur-Artal *et al.*, 2015). Approaches that do not extract features rely on variants of the Iterative Closest Point (ICP) algorithm to register dense depth data (Whelan *et al.*, 2015) or directly minimize photometric errors over all image pixels (Kerl *et al.*, 2013). These methods potentially exploit more information from the RGB and/or depth images, but they are computationally expensive and often require hardware acceleration on a GPGPU, which is not available on-board of most walking robots. In this paper, we evaluate systems that can run on a typical CPU, thus being deployable to on-board computers of walking robots. The four SLAM systems being investigated in this work are based on different state estimation principles and software architectures. Thus, we can draw conclusions as to the usefulness of these concepts for accurate localization of real-world legged robots. Moreover, we identify practical problems related to using an RGB-D camera mounted on a legged robot and investigate the relations between the robot motion strategy, and the trajectory estimation accuracy achieved by the on-board SLAM.

2 Related work

In the last fifteen years, we have witnessed a fast development in the area of robotic SLAM. Starting from various improvements of the classic Kalman filtering paradigm the focus in research has gradually moved to a more robust particle filtering approach (Durrant-Whyte and Bailey, 2006), and then non-linear optimization techniques have been adopted, leading to the graph-based approaches (Grisetti *et al.*, 2010). A graph-based SLAM system computes the sensor motion between consecutive image frames related to sensor poses, and then constructs a graph, whose vertices are those poses, whereas the motion-related constraints are treated as edges (Belter *et al.*, 2015a). This graph is optimized in order to minimize the discrepancy (errors) between the estimated and the measured poses of the vertices. If the robot re-visits an already known location, a loop closure introduces pose-to-pose constraints that correct drift of the trajectory. When this approach is applied to RGB-D images, the motion constraints between frames may be computed applying direct methods (Kerl *et al.*, 2013), or matching sparse features using visual descriptors (Endres *et al.*, 2014). Although pose-based SLAM with RGB-D data can produce reliable sensor trajectory estimates in real-time, this approach neglects a large part of the feature-to-pose constraints resulting from frequent observations of the same features (Strasdat *et al.*, 2012). These constraints are exploited in the bundle adjustment (BA) approach (Triggs *et al.*, 2000) to jointly optimize the sensor poses and feature positions. The BA-based architecture was introduced to real-time monocular SLAM by the breakthrough PTAM (Klein and Murray, 2007), and is still considered the gold standard in visual navigation for monocular (Mur-Artal *et al.*, 2015) and stereo (Mur-Artal and Tardos, 2016; Pire´ *et al.*, 2015) systems. The localization systems for micro aerial vehicles based on the structure of PTAM (Scherer and Zell, 2013) was an early attempt to adopt BA for RGB-D-based SLAM. The approach used in PUT SLAM is similar to BA, but whereas typical visual BA solutions minimize feature reprojection errors, in PUT SLAM, the errors in feature positions in 3-D space are minimized. This allows PUT SLAM to take advantage from advanced modeling of the spatial uncertainty of features (Belter *et al.*, 2016a).

Few papers tackle specifically the problem of self-localization or SLAM on a multi-legged walking robot. Robust localization of a hexapod robot in man-made environments using 2-D laser scans was demonstrated by Skrzypczynski (2012), whereas a 3-D time-of-flight camera´ was applied by Ronnau¨ *et al.* (2009) to localize a similar robot in rough terrain. Among visionbased systems an attempt to adopt monocular EKF-based SLAM for a hexapod robot was described by Schmidt and Kasinski (2010) with a preliminary demonstration of the resulting sys-´ tem, and a stereo-based approach to visual odometry in a hexapod was presented by Stelzer *et al.* (2012) with evaluation on rough terrain. Nonetheless, neither of these systems has been evaluated against ground truth robot/sensor trajectories or directly compared to other solutions. An evaluation of the PTAM system adopted for localization of a six-legged robot against simple ground truth trajectories was presented by Belter and Skrzypczynski (2013), resulting in the con-´ clusions as to the required localization accuracy for foothold planning. However, to the extent of the authors knowledge no broader study is available concerning the evaluation and comparison of RGB-D-based SLAM systems in the specific context of legged robot localization. Only very recently, Cˇ´ızek and Faigl (2016) presented a thorough evaluation of the RGB-D SLAM systemˇ (Endres *et al.*, 2014) on a six-legged crawler. This work was then extended to a comparison with the passive-stereo-based S-PTAM system (Fischer *et al.*, 2016). At the same time Belter *et al.* (2016c) compared four contemporary approaches to localization of a multi-legged robot with the RGB-D sensor. As the need for common ground for objective SLAM evaluation is acknowledged across the robotics community, we extend in this article the research of Belter *et al.* (2016c) and Cˇ´ızek and Faigl (2016) to provide a systematic experimental evaluation of theˇ representative SLAM architectures with RGB-D perception deployed on multi-legged robots.

3 SLAM architectures

In this section, we briefly present the four SLAM systems under investigation, focusing on the similarities and differences in their architectures. As the first one, we present PUT SLAM, which is the only one among the considered systems that was designed with the USAR mobile robotics applications in mind. Thus, this system is presented in more details, pointing out the elements of its architecture that are motivated by the specific application and the use of RGB-D sensing.

3.1 PUT SLAM

The Poznan University of Technology SLAM (PUT SLAM)[1] is based on the BA approach, that means it keeps a map of features and optimizes positions of these features jointly with the estimated sensor trajectory. Unlike the pose-based approach, which marginalizes the point features and constructs a graph whose vertices are only the sensor poses (Fig. 1a), PUT SLAM uses a graph with two types of vertices: $\mathbf{p}^f$ representing the features, and $\mathbf{p}^c$ representing the sensor poses (Fig. 1b). The i-th pose and the j-th feature are related by $\mathbf{t}_{i,j} \in \mathbb{R}^3$ constraint representing RGB-D measurement. Moreover, a pose-to-pose constraint $\mathbf{M}_{i,k} \in SE(3)$ resulting from the estimated sensor motion between the poses i and k can be introduced to the graph if the number of re-observed features belonging to the map is too small. Sensor poses are represented in the map global frame whereas the point features are represented relatively to the poses (keyframes) from which they have been observed for the first time. Considering this

[1] Open-source code is available at https://github.com/LRMPUT/PUTSLAM/tree/release

mathematical representation of the SLAM problem, the following function has to be minimized to optimize the map and trajectory:

$$\underset{\mathbf{p}}{\operatorname{argmin}}\, F = \sum_{i=1}^{n}\sum_{j=1}^{m} \mathbf{e}_{f(i,j)}^{T} \mathbf{\Omega}^{t}_{i,j} \mathbf{e}_{f(i,j)} + \sum_{i,k=1, i\neq k}^{n} \mathbf{e}_{c(i,k)}^{T} \mathbf{\Omega}^{M}_{i,k} \mathbf{e}_{c(i,k)}, \qquad (1)$$

where $\mathbf{e}_{f(i,j)} = e(\mathbf{p}^c_i, \mathbf{p}^f_j, \mathbf{t}_{i,j})$ and $\mathbf{e}_{c(i,k)} = e(\mathbf{p}^c_i, \mathbf{p}^c_k, \mathbf{M}_{i,k})$ are error functions for the feature-topose and pose-to-pose constraints, respectively. The accuracy of each feature-to-pose or pose-topose constraint is represented by its respective information matrix $\mathbf{\Omega}^t_{i,j}$ or $\mathbf{\Omega}^M_{i,j}$. These matrices are obtained from the spatial uncertainty model of features (Belter *et al.*, 2016a) or can be set to identity matrices as in Endres *et al.* (2014). The graph representation is held in the back-end together with the map structure, which contains point features augmented by their visual ORB descriptors. The back-end uses the g^{2}o graph optimization library (Kummerle¨ *et al.*, 2011) to solve (1).

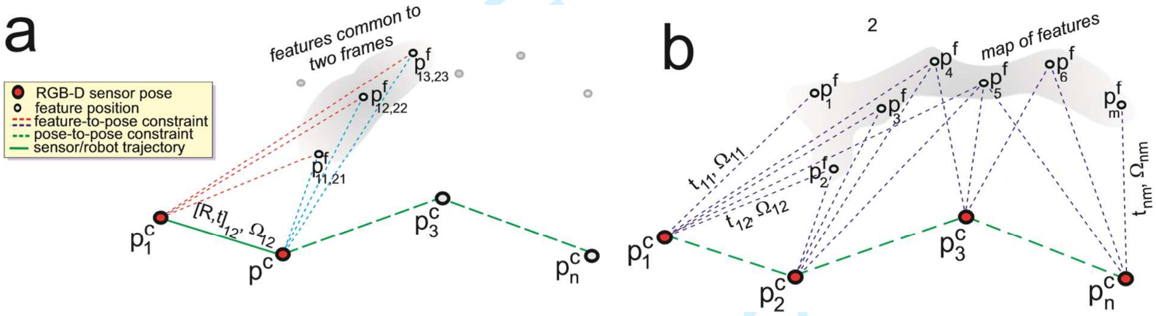

Figure 1: Graph structure of pose-based SLAM (a) and BA-based SLAM (b)

PUT SLAM fully exploits the RGB-D data obtained from the Kinect or Xtion sensor to quickly extract 3-D point features and estimate their spatial uncertainty (Belter *et al.*, 2016a). Processing of the incoming RGB-D frames takes place in the front-end, which also tracks the sensor against the map of features. While other visual SLAM systems use a mathematical model of the sensor motion (Klein and Murray, 2007; Mur-Artal *et al.*, 2015; Pire *et al.*, 2015), in PUT SLAM, the initial guess of sensor displacement is computed by a fast visual odometry pipeline based on the photometric Lucas-Kanade keypoint tracker (Belter *et al.*, 2015a). This helps to handle the rapid and unpredictable motion of the robot carrying the sensor. The Umeyama algorithm (Umeyama, 1991) is applied to compute the 3-D-to-3-D transformation between two consecutive frames from the set of tracked features augmented with the corresponding depth data. RANSAC is employed to ensure robustness to outliers. The feature keypoints extracted by visual odometry are then augmented by ORB descriptors and used for matching between the features from an incoming frame and the map. The binary ORB detector/descriptor was chosen mainly because of its computational efficiency and good performance in the context of visual navigation (Schmidt *et al.*, 2013). Moreover, the corner-like ORB keypoints are well suitable for photometric tracking, unlike their blob-like counterparts, produced for example by SURF. Knowing the initial guess of the sensor pose enables to consider as matching candidates only the map features located in a small neighborhood of the predicted features in the image plane, which increases robustness. The matching features establish constraints between the current sensor pose and the features already existing in the map (generated from the previous, possibly much older, frames). Owing to this mechanism PUT SLAM handles local metric loop closures implicitly, without a need to detect the re-visited locations by their appearance.

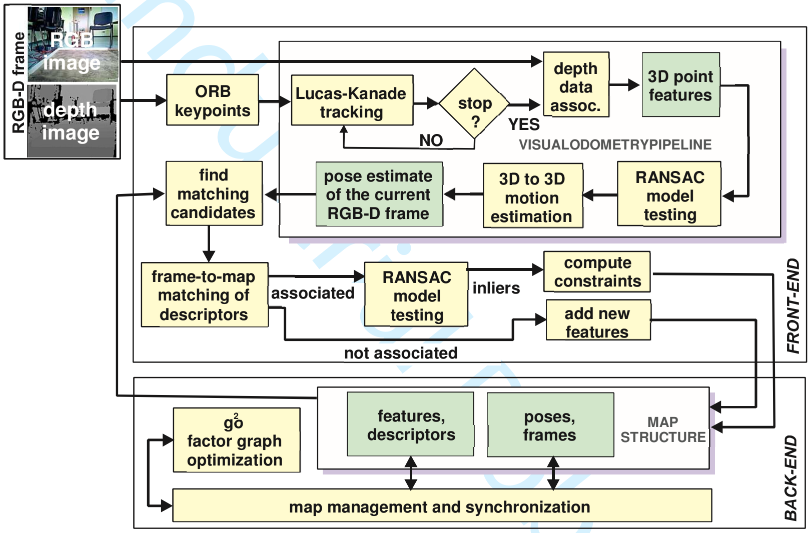

Figure 2: Block scheme of PUT SLAM software

The software architecture of PUT SLAM is presented in Fig. 2. The front-end and the backend work asynchronously in separate threads. They get synchronized on specific events, such as insertion of new constraints or prediction of map features for matching. Owing to this software architecture PUT SLAM efficiently uses a multi-core CPU and runs in real-time without a GPGPU.

3.2 RGB-D SLAM v2

This is the latest implementation[2] of the algorithm presented in details by Endres *et al.* (2014), which follows the pose-based SLAM approach without storing visual features in the map. The RGB-D SLAM v2 architecture is also divided into a front-end and a back-end that run in parallel. The front-end implements frame-to-frame RGB-D visual odometry employing 3-D-to-3-D matching of point features with visual descriptors (ORB, SURF or SIFT detector/descriptor pairs can be configured). Next, the front-end attempts to detect loop

[2] Source code is available at https://github.com/felixendres/rgbdslam _v2/tree/indigo

closures to decrease trajectory drift. The last RGB-D frame is compared to previous keyframes using a heuristic strategy of searching over the past keyframes. To determine possible loop closures the RGB-D SLAM v2 applies direct frame-to-frame matching of local descriptors using RANSAC, without recoursing to the appearance-based recognition of locations. The current frame is marked as a new keyframe when it cannot be matched to any keyframe stored by the back-end.

Finally, the graph of connected keyframes is optimized by using the g^{2}o library, which is also employed in PUT SLAM. All experiments presented in this paper were performed using the SURF-based variant of RGB-D SLAM v2, as results presented in both Belter *et al.* (2016c) and Cˇ´ızek and Faigl (2016) suggest that with SURF this system yieldsˇ best results even without the GPGPU acceleration. In the g^{2}o back-end the PCG solver is used, and the loop closure parameters are set as in Fischer *et al.* (2016).

3.3 CCNY RGBD

The CCNY RGBD algorithm was introduced by Dryanovski *et al.* (2013), and its ROS-based implementation is available at GitHub[3]. The algorithm employs depth and RGB data in a featurebased framework with a persistent map of features. However, it differs remarkably from the PUT SLAM architecture. Although CCNY RGBD can be configured with one of four keypoint detectors: ORB, SURF, STAR or GFT, in the case of ORB and SURF this system does not use the descriptor part. Instead, the sparse ICP method is used to align the detected keypoints to the map. A Mahalanobis-like distance function is used in the alignment algorithm to take into account the spatial uncertainty of point features. Then, the positions of features in the map are updated using the Kalman filter, with a Kinect-specific feature uncertainty model. If the maximum size of the map is exceeded, the oldest features are replaced with the newest ones.

The CCNY RGBD does not use any loop closing procedure based on place recognition, thus it can detect and close only small-scale loops by feature re-observations. However, there exists a version, which can perform map optimization using g^{2}o, but the optimization works off-line, on the final map and trajectory produced by the system. Therefore, it is not used in our evaluation. For the experiments, the CCNY RGBD is configured with the ORB detector, used also in PUT SLAM and ORB-SLAM2, and the maximum map size is set to 10000 features.

3.4 ORB-SLAM2

The very recent ORB-SLAM2 (Mur-Artal and Tardos, 2016) is the second version of the open-´ source[4] visual SLAM system presented in detail by Mur-Artal *et al.* (2015). Initially, it was created as monocular SLAM employing point features and the BA approach to optimize both

[3] https://github.com/ccny-ros-pkg/ccny rgbd tools

[4] Available at https://github.com/raulmur/ORB SLAM2

the camera trajectory and the map of features. The second version can use either passive stereo or RGB-D data, thus avoiding the problem of proper initialization of the map from monocular camera images when no depth information is available. The graph of constraints contains both sensors poses and feature positions as vertices, and it is optimized using the g^{2}o library, as in PUT SLAM.

Very similar is also the general architecture of both systems. ORB-SLAM2 is divided into the front-end processing incoming data, and the back-end storing and optimizing the map. The main differences between these two architectures are due to the fact that ORB-SLAM2 retains all the properties of a monocular SLAM system, being able to process keypoints from RGB images that lack depth information. This allows ORB-SLAM2 to handle also features located outside the depth range of the RGB-D sensor. Another important difference is related to the definition of the error function in the optimization – ORB-SLAM2 always minimizes the reprojection error of features onto RGB images, even working with RGB-D data that produce readily available 3-D points. This solution, standard for the BA in computer vision, makes the optimization vulnerable to errors that affect the location of pixels in RGB images, such as motion blur and rolling shutter. On the other hand, relying on reprojection errors makes it possible to use features that do not have proper depth values and increases robustness to range measurement errors and artifacts in depth images. In the context of walking robot, the important difference is also in the simple constant velocity motion model used by ORB-SLAM2 to predict the sensor pose in tracking.

ORB-SLAM2 uses multi-scale ORB descriptors and divides the point features into two categories: close and far, using an inverse depth parametrization for the latter. RGB-D depth measurements are used only for close features, whereas they are ignored for the far ones. The ORB features are used throughout the system – in sensor tracking, matching the current perception to the map (establishing constraints), and in closing the loops. ORB-SLAM2 implements appearance-based loop closure detection re-using the same ORB features. This allows closing a loop of any size, not only local metric loops, even though this process may not run in real-time for very large loops.

4 Evaluation Methodology

4.1 Motivation and evaluation metrics

Data sets such as the TUM RGB-D Benchmark (Sturm *et al.*, 2012) facilitate benchmarking of SLAM systems and enable comparison between various approaches. However, the data sets obtained either using a handheld Kinect/Xtion sensor or a sensor of the same class attached to a wheeled robot do not allow us to evaluate robustness of the SLAM systems to problems characteristic to walking robots. Legged locomotion is discrete, which gives rise to uncontrolled oscillations of the robot attitude (pitch and roll angles of the trunk in particular). Walking robots are also characterized by a high level of vibrations, particularly if low-cost servomotors are used in the legs. All that introduces significant motion blur into the images acquired in motion. Additionally, the field of view of an RGB-D sensor mounted on a small or mid-size walking robot is usually far from being optimal for SLAM.

Therefore, we have performed a series of experiments on two different six-legged robots: the in-house developed Messor II (Belter and Walas, 2014) and the commercially available PhantomX platform. In order to facilitate comparison of the SLAM architectures we demonstrate the trajectory estimation accuracy with quantitative results, applying the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) metrics introduced by Sturm *et al.* (2012). The ATE measures the distance between corresponding points of the estimated and ground truth trajectories, while the RPE metric is correlated to the local drift of the trajectory. Computing these metrics for a given SLAM-estimated trajectory of the sensor $\mathbf{T} = \{\mathbf{T}_1,\mathbf{T}_2,...,\mathbf{T}_n\} \in \mathrm{SE}(3)$ we assume that there exists a reference trajectory $\mathbf{T}^{\mathrm{gt}}$, which is synchronized with $\mathbf{T}$ and contains the same number of n sensor poses $\mathbf{T}^{\mathrm{gt}} = \{\mathbf{T}_1^{\mathrm{gt}}, \mathbf{T}_2^{\mathrm{gt}}, \ldots, \mathbf{T}_n^{\mathrm{gt}}\} \in \mathrm{SE}(3)$, where $\mathbf{T}_i$ and $\mathbf{T}^{\mathrm{gt}}_i$ are sensor poses in 3-D for the *i*-th time stamp (or frame in the RGB-D sequence) expressed by 4×4 homogeneous matrices. The $\mathbf{T}^{\mathrm{gt}}$ trajectory is obtained from an external ground truth acquisition system. Both trajectories are synchronized by frame numbers or by alignment of the time stamps, depending on the motion capture system that is employed. The RPE metrics for *i*-th frame is given by the equation:

$$\mathbf{E}_i^{\mathrm{RPE}} = \left((\mathbf{T}_i^{\mathrm{gt}})^{-1}\mathbf{T}_{i+1}^{\mathrm{gt}}\right)^{-1}\left(\mathbf{T}_i^{-1}\mathbf{T}_{i+1}\right). \qquad (2)$$

Taking the translational or rotational part of the homogeneous matrix $\mathbf{E}^{\mathrm{RPE}}_i$ and computing the Euclidean norm or Euler angle we obtain the relative translational $\mathrm{RPE}_t(i)$ or rotational $\mathrm{RPE}_\theta(i)$ error at *i*-th frame, respectively. The RPE_t and RPE_θ metrics for the whole recovered trajectory are computed from the Root Mean Square Error (RMSE) for all nodes of $\mathbf{T}$ and $\mathbf{T}^{\mathrm{gt}}$. As ATE compares absolute distances between nodes of the synchronized trajectories, it is necessary to align the trajectories by finding a rotation between these two rigid sets of points that minimizes the distance between them (Sturm *et al.*, 2012). Then, the ATE metrics for the *i*-th frame is computed as:

$$\mathbf{E}_{\mathrm{ATE}_i} = \left(\mathbf{T}_i^{\mathrm{gt}}\right)^{-1}\mathbf{T}_i \qquad (3)$$

and the ATE value for the whole trajectory is computed from the RMSE of (3) for all nodes. Besides, the standard deviation of the absolute trajectory error σ_{ATE} is computed.

4.2 Experimental set-up with Messor II

The Messor II robot is a hexapod with 18 control degrees of freedom in its legs (Fig. 3). It uses the Asus Xtion Pro Live RGB-D sensor based on the PrimeSense structured light depth measuring technology. On the Messor II, this sensor is mounted at the height of 40 cm above the ground, (assuming the default posture of the robot) and slightly tilted down. Although the Messor II has an Inertial Measurements Unit (IMU) that can estimate its attitude with respect to the gravity vector, in the present implementation of our SLAM system these measurements are not used. This ensures a fair comparison between systems that were not designed to accommodate IMU measurements. All experiments with the Messor II were performed on a sand-covered terrain mockup of the size 2×2 m (Fig. 4a). Some objects have been placed around the mockup to make the environment richer in features. The configuration of these objects was slightly different for each experiment.

The ground truth trajectories were collected using a ceiling-mounted system of cameras. This system consists of five high-resolution Basler acA1600 cameras equipped with low-distortion lenses, but for the presented experiments only the central camera was used (see Fig. 5a, the used camera is pointed by the arrow), because of a limited area of the terrain mockup traversed by the Messor II (Fig. 5b). Employing a single camera simplified the registration procedure, as mutual calibration of the co-operating cameras was not necessary (Schmidt *et al.*, 2014). We exactly synchronized the RGB-D frame acquisition rate of the Xtion sensor on the robot with the frame rate of the overhead camera to avoid evaluation errors due to trajectory interpolation. The synchronized frame acquisition rate 15 Hz is limited by the parameters of the overhead camera.

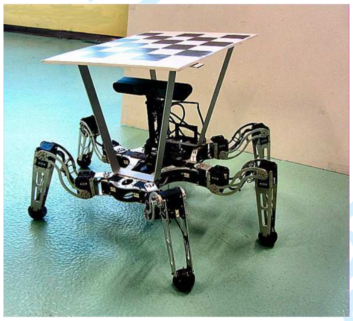

Figure 3: Messor II, an autonomous six-legged robot with the Asus Xtion Pro Live RGB-D sensor and the chessboard marker for ground truth acquisition

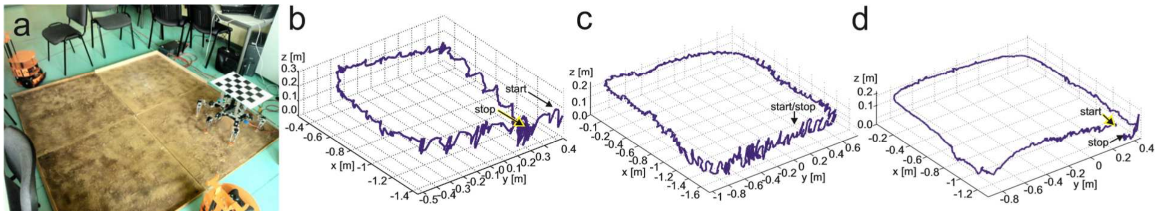

Figure 4: Experiments with the Messor II. Experimental set-up (a), and the ground truth trajectories: fast tripod gait (b), slower tripod gait (c), slow crawl gait (d)

The software of the multi-camera system detects and localizes a chessboard marker rigidly attached to the Messor II robot. The known size pattern was extracted from images and its pose was computed with respect to the overhead camera. Then, the pose of the Xtion sensor with respect to the overhead camera was computed using the rigid transformation obtained by the calibration. The rigid transformation ${}^{M}\mathbf{T}_{X}$ between the Xtion coordinate system X, and the coordinates of the chessboard marker M were estimated applying the calibration algorithm from (Schmidt *et al.*, 2014). Images of an external calibration pattern observed by the on-board Xtion's RGB camera and images registered by the overhead camera observing both the external pattern and the marker attached to the robot were used for calibration (Fig. 5c).

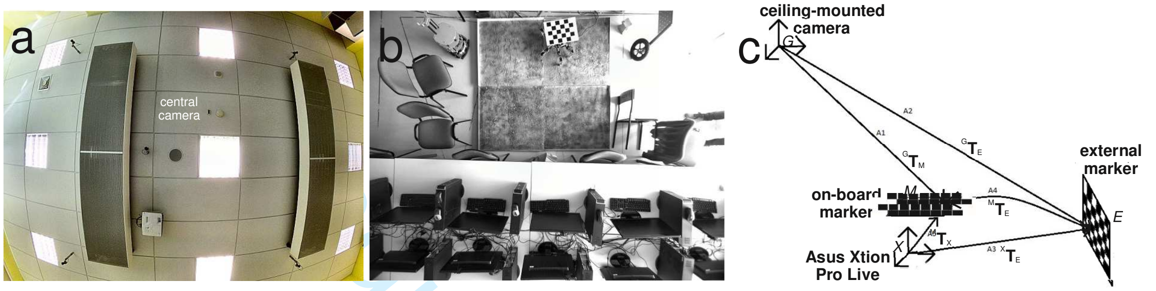

Figure 5: Ground truth acquisition for the Messor II robot: ceiling-mounted cameras (a), experimental set-up seen by the overhead camera (b), and the calibration concept (c)

4.3 Experimental set-up with PhantomX

The second experiment is focused on rough terrain conditions and has been performed with the PhantomX hexapod walking robot (Fig. 6). The robot has six legs attached to the trunk that hosts all sensors and the control unit. Each leg is composed of three intelligent servo motors. Estimation of the torque at the joint is utilized in the adaptive motion control (Mrva and Faigl, 2015) that enables the robot to negotiate the terrain the robot is crawling on. The used adaptive motion gait is based on a regular tripod gait enhanced by the ability to sense the contact of individual legs with the ground by utilizing the servo drives feedback only (Mrva and Faigl, 2015). The estimated tactile sensing is based on the servo position error that allows stopping the movement of the leg avoiding high torque values at the joints. The motion is divided into displacement of the individual legs with fixed body that is followed by a body movement with fixed legs. Therefore, the RGB-D sensor movements are smooth but periodic. This not only enables to traverse rough terrain but provides also smooth motion without significant impacts when the legs hit the ground. Due to the used adaptive motion gait, the speed of the robot is approximately 0.014 m/s and it is almost constant regardless the terrain being traversed.

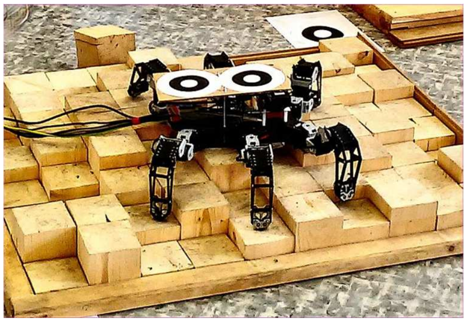

Figure 6: PhantomX, a six-legged robot with WhyCon markers for ground truth acquisition

The only sensor used for SLAM systems evaluation is the Asus Xtion Pro Live RGB-D attached to the robot body about 20 cm above the ground. The used height prohibits the depth sensor to observe the terrain directly in front of the robot on the ground level, but it reduces shaking of the camera during locomotion through the rough terrain. Besides, the XSens MTi-30 AHRS inertial measurement unit has been attached to the robot body to provide a ground truth orientation of the robot. This unit is, however, not used by the robot control algorithm and the evaluated SLAM systems.

The experimental test track consists of a square path of approximately 9 m length involving a plain ground, a hill of irregularly height wooden blocks, a set of stairs, and an inclined ramp. The test track is visualized in Fig. 7a. Each of the particular terrain types represents a different challenge for the hexapod platform and also the SLAM system. The test track is located in a lab environment which has not been distinctly modified to be richer in features, except a few boxes placed in the first two corners of the trajectory to provide the algorithms with features both in the near and distant field of view. The hill of irregularly height wooden blocks has been designed to evaluate the SLAM methods on an uneven terrain where angular rotation of the camera can occur with a high frequency. The stairs represent a challenge where the camera pose changes abruptly and randomly because of slippage. During the descent, the forward-looking sensor is inclined to the ground and only limited number of features in a close distance is visible. The individual turns represent another significant challenge for the SLAM systems due to the motion blur in the acquired images.

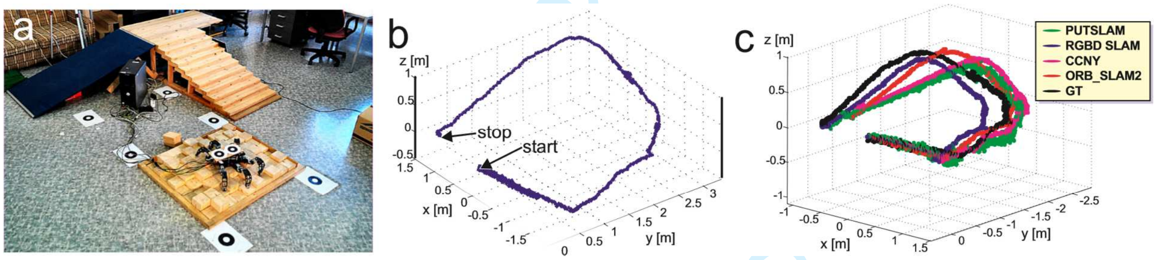

Figure 7: Experiment with the PhantomX. Experimental set-up (a), ground truth trajectory (b), and visualization of the recovered trajectories (c)

The 6 DOF ground truth for the experiment has been obtained by visual tracking of the robot position using WhyCon system (Krajnik *et al.*, 2014) that is accompanied by the orientation of the robot provided by the precise XSens MTi-30 AHRS unit. The visual localization system tracks circular patterns rigidly mounted to the robot by a multicamera setup to provide a precise 3 DOF position of the robot. Due to the size of the test track, two cameras have to be employed to cover the whole experimental arena with the achieved accuracy of about 2 cm, which has been considered as sufficient for this research. The tracked position of the robot by the individual cameras have been merged into a single coordinate frame using four static markers on the ground (see Fig. 7a). The acquisition rates for the utilized Asus Xtion Pro

RGB-D sensor, the WhyCon reference system, and the AHRS unit are restricted to 10 frames per second, 6 frames per second, and 400 samples per second, respectively. The synchronization of the RGB-D data with the ground truth data is based on timestamping and interpolation similarly as it has been used by Sturm *et al.* (2012). However, the slow average velocity of the robot locomotion does not cause a significant error of the interpolation.

5 Experiments and results

In the first experiment, three runs were performed with the Messor II. The shape of the test trajectories resemble rectangles of the size similar to the size of the mockup (see Fig. 4b, c, and d). The runs differ mainly in the velocity of the robot, and the gait type being used. In the first two runs, the Messor II uses its regular tripod gait. The tripod gait is the fastest statically stable gait of this walking robot. The robot was controlled by a human tele-operator using a joystick. Thus, the step length wasn't constant, and the effective velocity of the robot fluctuated slightly along the path. For the first experiment, the velocity was 0.15 m/s, which is 95% of the maximal value. In the second run, the velocity was decreased to 45% of the maximal one, resulting in the velocity value of 0.09 m/s. In the third run, the gait was changed to a slower crawl gait. During the crawl gait, the single leg of the robot is in the swing phase at once. This resulted in a velocity of about 0.05 m/s.

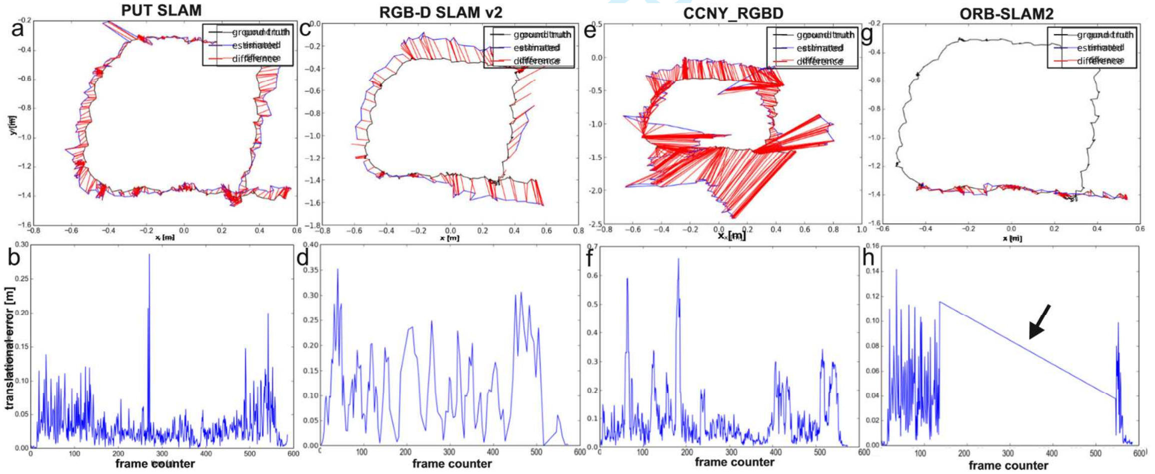

Figure 8: SLAM evaluation results for the Messor II robot walking at 95% of its maximal speed: estimated trajectories with ATE (a,c,e,g), and translational RPE plots (b,d,f,h)

Figure 8 shows the plots of ATE and RPE metrics generated by the (from left to right) PUT SLAM, RGB-D SLAM v2, CCNY RGBD, and ORB-SLAM2 evaluated on the sequence of frames obtained in the first run (`messor2_1`, Fig. 4b). The accuracy of trajectory estimation is rather low for all the SLAM systems. Albeit PUT SLAM produced the smallest absolute (Fig. 8a) and relative (Fig. 8b) errors, the RMSE ATE of about 9.5 cm is too large if we want to register the depth measurements into a global terrain model (Belter and Skrzypczynski, 2013).´ The RGB-D SLAM v2 was also capable of accomplishing the trajectory reconstruction, but its RMSE ATE is about 15 cm (Fig. 8c). Note that in spite of yielding a reasonable global trajectory estimate RGB-D SLAM v2 generated much larger relative errors (Fig. 8d). The CCNY RGBD was not able to compensate some large pose drifts that appeared along the trajectory, and the obtained trajectory estimate is useless (Fig. 8e). Surprisingly, the ORB-SLAM2, which is known for providing extremely precise run trajectory estimates on several benchmarks (Mur-Artal and Tardos, 2016), failed on the´ `messor2_1` sequence (Fig. 8g). It lost tracking of the sensor at the end of the first leg of the trajectory, apparently during a sharp turn of the robot. The system was unsuccessful in trying to re-localize after this failure (see the arrow in Fig. 8h). In this run the main reason for large sensor pose errors or even failure of the evaluated SLAM methods seems to be the excessive amount of motion blur appearing in the collected RGB-D frames. This made detection and description of point features unreliable, and in the case of ORB-SLAM2 perhaps affected also the reprojection error computations.

In the second run, denoted the `messor2_2` sequence (Fig. 4c), the speed of the robot was reduced to 45% of the maximal value, and the robot's trunk vibrations had smaller amplitude. In this case, PUT SLAM estimates the trajectory with a better, but still not acceptable precision (Fig. 9a). There was no noticeable improvement of ATE and RPE metrics for the RGB-D SLAM v2 and CCNY RGBD systems, although the CCNY RGBD trajectory is qualitatively better (Fig. 9e). Also in this run, ORB-SLAM2 was unable to complete the run, but it managed to process

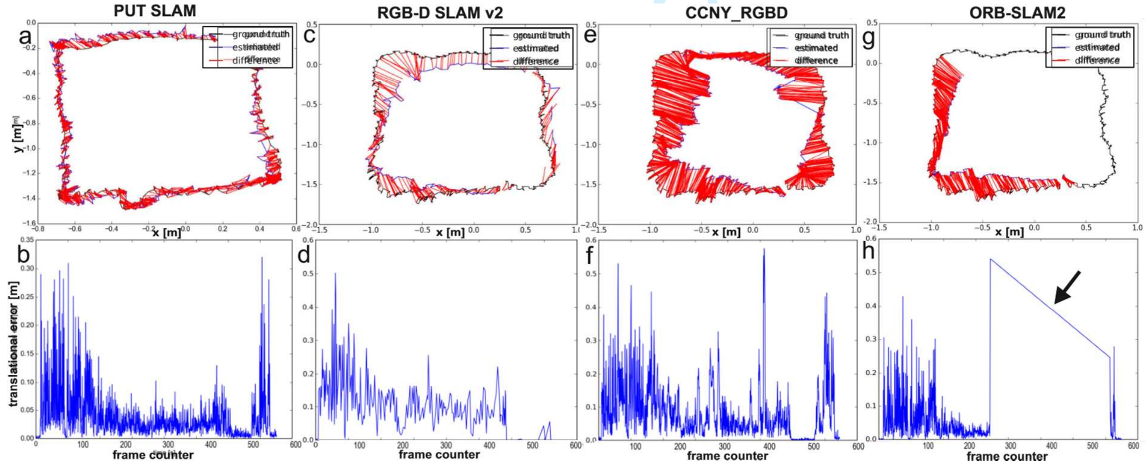

Figure 9: SLAM evaluation results for the Messor II robot walking at 45% of its maximal speed: estimated trajectories with ATE (a,c,e,g), and translational RPE plots (b,d,f,h)

about a half of the sequence (Fig. 9g). The reasons for the failure were very similar – the inability to match enough features when the robot was almost turning on the spot at the corner of the trajectory, followed by an unsuccesful re-localization (see the arrow in Fig. 9h).

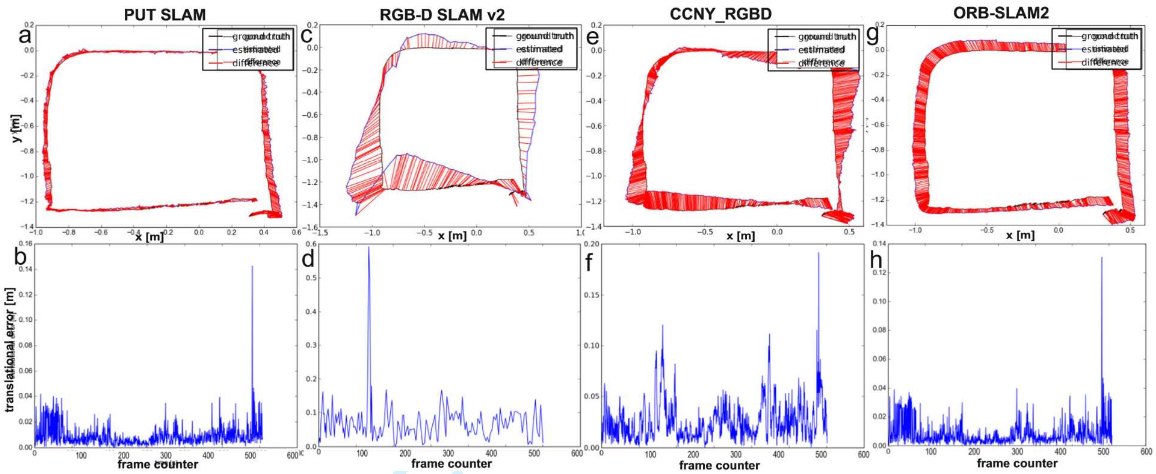

Figure 10: SLAM evaluation results for the Messor II robot using crawl gait: estimated trajectories with ATE (a,c,e,g), and translational RPE plots (b,d,f,h)

In the third run (messor2_3, Fig. 4d), the Messor II motion was slower than the previous runs, because of the crawl-type gait being used. Therefore, all RGB-D frames were acquired when the robot was supported by at least five legs, avoiding vibrations due to the momentarily overload of the servos, and significantly reducing the motion blur. The PUT SLAM system was able to recover a smooth trajectory with RMSE ATE of about 5 cm (Fig. 10a). This accuracy should be satisfactory for terrain mapping and motion planning. Surprisingly, there was no improvement in the trajectory estimation by RGB-D SLAM v2 (Fig. 10c), although RPE values are reduced (Fig. 10d). Conversely, the results of CCNY RGBD improved (Fig. 10e and 10f). This sequence was the first one on which ORB-SLAM2 recovered the full trajectory. As both RMSE ATE and RPE values were small, we can conclude that the motion blur on RGB images (practically not present in this sequence) combined with fast rotations of the robot were the main failure causes of ORB-SLAM2 in the first experiment. Table 1 summarizes the quantitative results of the first experiment for all runs and all the SLAM systems.

Table 1: ATE and RPE values obtained by the tested systems for the messor2_ sequences

RGB-D sequence and SLAM architecture	ATE [m]	σ_{ATE} [m]	RPE$_t$ [m]	RPE$_\theta$ [deg]
PUT SLAM messor2_1	0.095	0.052	0.054	4.96
RGB-D SLAM v2 messor2_1	0.155	0.059	0.138	15.93
CCNY RGBD messor2_1	0.468	0.262	0.140	10.02
ORB-SLAM2 messor2_1	—	—	—	—
PUT SLAM messor2_2	0.069	0.033	0.041	5.66
RGB-D SLAM v2 messor2_2	0.210	0.082	0.139	10.54
CCNY RGBD messor2_2	0.307	0.107	0.118	6.39

ORB-SLAM2 messor22	—	—	—	—
PUT SLAM messor2_3	0.053	0.025	0.012	5.42
RGB-D SLAM v2 messor2_3	0.147	0.062	0.094	° ° ° 11.04°
CCNY RGBD messor2_3	0.102	0.044	0.032	6.37
ORB-SLAM2 messor2_3	0.085	0.019	0.012	5.43

The second experiment conducted with the PhantomX hexapod focuses on the evaluation of the SLAM systems in the rough terrain scenario. During this experiment, the robot has been remotely guided by a human operator along the whole path. Altogether five experimental runs have been performed on the test track from which we have selected for evaluation the run where the robot did not finish the whole path, and therefore, did not close the loop (Fig. 7b). Due to the adaptive gait, the speed and motion type of the robot were similar in all runs. Thus, we have chosen a sequence, which is interesting because it best evaluates the ability of a SLAM system to work while exploring an unknown environment, which is a typical scenario in USAR missions. Moreover, it was also the most challenging trajectory in terms of camera shaking and abrupt motions.

The evaluation results are summarized in Tab. 2 and depicted in Fig. 11 for (from left to right) PUT SLAM, RGB-D SLAM v2, CCNY RGBD, and ORB-SLAM2. The constant-velocity smooth gait directly affects the resulting trajectory and the SLAM outcome as well, because the particular type of terrain ceases to affect the precision of the SLAM system (Cˇ´ızek andˇ Faigl, 2016). The large values of ATE and RPE are mostly caused by inability of an algorithm to track the features in specific parts of the trajectory which are usually related to fast and abrupt motions when the camera pose changes unexpectedly or the parts of the sequence which contain a significant motion blur. Therefore, the recovered trajectories are similar for all the evaluated solutions (Fig. 7c). In spite of these similarities specific problems and failures can be observed for particular architectures.

Table 2: ATE and RPE values obtained by the tested systems for the phantomx sequence

SLAM architecture	ATE [m]	σ_{ATE} [m]	RPE$_t$ [m]	RPE$_\theta$ [deg]
PUT SLAM	0.171	0.074	0.044	11.75
RGB-D SLAM v2	0.193	0.036	0.021	° ° ° 11.72°
CCNY _RGBD	0.183	0.061	0.023	11.85
ORB-SLAM2	0.164	0.058	0.021	11.71

Although ORB-SLAM2 scored first with respect to the RMSE ATE value (see Tab. 2), it was unable to complete the trajectory to the very end. It loses tracking close to the end of the sequence. The failure is caused by the rapid orientation change of the robot with additional vibrations, characteristic to a legged robot. The inability to track the sensors pose due to motion blur is a typical problem for visual SLAM systems, which in ORB-SLAM2 manifests itself also when RGB-D data are used. In this sequence, ORB-SLAM2 loses tracking also earlier on the trajectory (arrow no. 1 in Fig. 11g), but is able to recover when the robot takes a short pause. But toward the end of the trajectory, it loses the tracking once again and this time does not recover (arrow no. 2 in Fig. 11g)[5]. This is a result of not adding new map features fast enough, as ORB-SLAM2 usually localizes itself only on well-established map features. The ORB-SLAM2 system once lost, it always tries to re-localize itself against the already built map, which might be also considered a disadvantage of this architecture, because ORB-SLAM2 does not restart the mapping procedure if the tracking is lost, and the robot is exploring a new environment (as in our experiment), rendering the SLAM module useless. The map built by the ORB-SLAM2 till the moment it got lost contains 7610 point features.

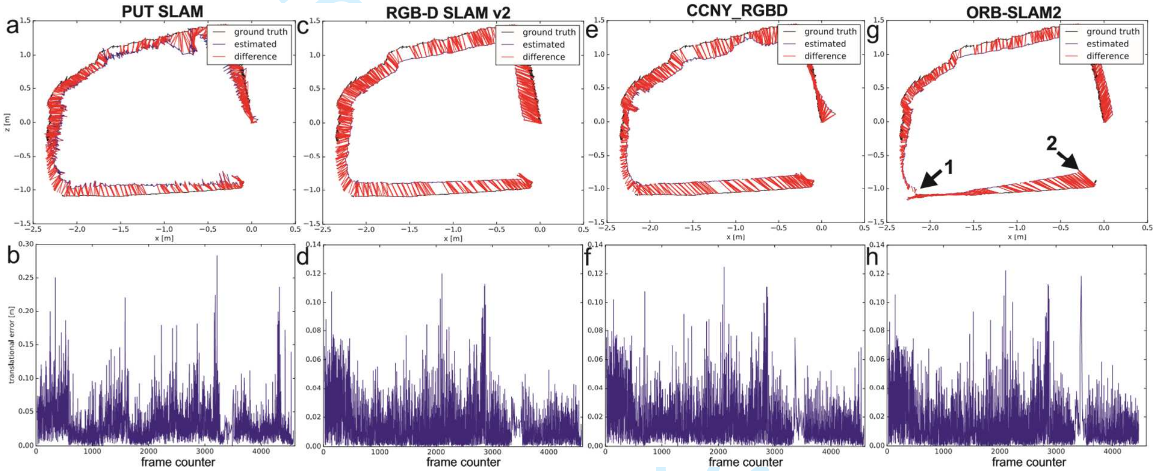

Figure 11: SLAM evaluation results for the PhantomX robot in rough terrain: estimated trajectories with ATE (a,c,e,g), and translational RPE plots (b,d,f,h)

The PUT SLAM scored second in the RMSE ATE category. This system was able to optimize the whole recovered trajectory and map yielding a reasonable absolute error as for an openloop sequence, but at the same time, it produced larger transitional RPE errors, which can be problematic for relative localization, e.g., when planning footholds. We attribute this problem to the very slow motion of the PhantomX robot, which combined with the 10 Hz frame rate of the RGB-D sensor resulted in very small distances between consecutive frames. Whereas this is typically good for vision-based SLAM systems (Klein and Murray, 2007) in PUT SLAM it may lead to accumulating errors in Lucas-Kanade feature tracking and increased the uncertainty of the depth measurements for the observed features. The PUT SLAM had the largest relative error when the robot was turning at the top of the stairways. At this moment almost all of the features were located pretty far, which increased the Xtion depth measurements errors. The PUT SLAM is more prone to this type of problems than systems relying more on the visual information. The PUT SLAM successfully managed rapid orientation changes of the robot at locations rich in close features, e.g., at the first turn of the trajectory. In such situations, when only a limited number of features can be re-observed due to

[5] See also the video https://www.youtube.com/watch?v=cgYo6ISxPZk

fast changes of the field of view, this system quickly adds new features to the map (see Fig. 12 around frame no. 1000), which prevents it from getting lost. However, the keyframe-culling mechanism in the back-end allows PUT SLAM to delete the excessive features with a minimal number of re-observations. The map grows with time, reaching 8330 features at the end of the sequence, but this is caused by the exploratory nature of the trajectory.

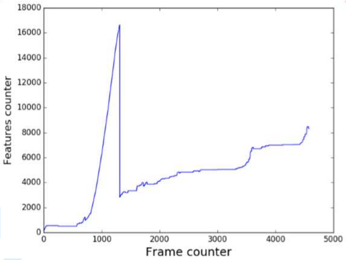

Figure 12: Evolution of the number of map features in PUT SLAM for the `phantomx` sequence

The relatively simple CCNY RGBD system performed surprisingly well in this experiment. This system is less prone to problems with fast rotations. However, the RMSE ATE value is larger than for the BA-based architectures. It should be noted that while the BA-based systems expand their maps gradually along the trajectory, reaching about 8000 features, the CCNY RGBD local map has a constant size, and the oldest features are replaced by the most recent. Perhaps this simple strategy makes the local metric loop closures less efficient, as some useful features may disappear with time, eventually leading to a drift of the estimated trajectory.

Similarly, the RGB-D SLAM v2 produced a smooth trajectory, which in spite of the reasonable translational RPE has the largest RMSE ATE among all the tested systems. This suggests that the pose-based approach is less efficient in the integration of a large amount of information provided by a feature-based front-end. The RGB-D SLAM v2 system was already evaluated thoroughly on very similar sequences from the same robot (Cˇ´ızek and Faigl, 2016). This evalu-ˇ ation revealed that the performance of this architecture to a large extent depends on the chosen configuration and the detector/descriptor type in particular. As Cˇ´ızek and Faigl (2016) conclude,ˇ the performance of RGB-D SLAM v2 with the ORB features is not satisfactory because of their poor discriminability. Therefore, we configured this SLAM system with the much slower to compute SURF features. This yielded results comparable to other systems under investigation, but at the cost of making RGB-D SLAM v2 slower than these systems.

In the second experiment the frame rate of the recorded RGB-D sequence was 10 Hz, and all the evaluated systems were able to work in real-time on a standard i5 PC computer without GPGPU acceleration. The Ubuntu Linux was used as the operating system, with the ROS framework for the PUT SLAM, RGB-D SLAM v2, and CCNY RGBD systems. We have noted that ORB-SLAM2 took only 55% to 65% of the processor time, being thus the most efficient SLAM system in our tests. On the other hand, we have observed that the RGB-D SLAM v2 slowed down with the growing amount of keyframes, as the optimization took more time. In the experiment with the Messor II sequences, when the frame rate was higher, the RGB-D SLAM v2 was not able to process the incoming frames in real-time.

6 Conclusions

The experiments presented in this article revealed that the predominant problem characteristic to the legged robots as platforms for SLAM with RGB-D perception are the abrupt and unpredictable sensor motions, as well as oscillations and vibrations, which corrupt the images captured in-motion. In the Messor II experiment substantially reduced trajectory estimation errors were observed if the RGB-D frames were collected when the robot was supported by five legs. This suggests that the perception process should be tightly coupled with the gait control algorithm. The PhantomX experiment results support this conclusion, as the used adaptive gait, albeit slow, allowed all the evaluated SLAM systems to reconstruct acceptable trajectories, in spite of the more challenging terrain.

The two BA-based systems produced best results, which is attributed to the use of a persistent map of visual features, which enables to establish a large number of constraints for the estimated trajectory. However, both evaluated BA-based systems have their distinctive features. The PUT SLAM utilizes a very fast visual odometry pipeline, which allows tracking the sensor even during fast rotations of the robot. The BA optimization based on the Euclidean error involving depth measurements seems to help this system to overcome some problems with motion blur and rolling shutter in the acquired images. The ORB-SLAM2 failed altogether on the two sequences with highly blurred images in the first experiment. But on the other hand, the ORBSLAM2 architecture having its roots in a pure visual SLAM algorithm handles more efficiently features located at various distances, also those detected beyond the effective range of the depth camera. Also the ORB feature management and optimization strategy in ORB-SLAM2 appear to be more efficient than their counterparts in PUT SLAM.

The third architecture implementing a persistent map of features – CCNY RGBD also demonstrated much accuracy boost with the improving quality of images in the first experiment. However, the Kalman filter used in CCNY RGBD was probably unable to handle the highly uncertain feature locations obtained from blurred images, whereas this was possible using BA in PUT SLAM. The RGB-D SLAM v2, which also uses the g^{2}o optimization engine, as the BA-based systems, but in the pose-based SLAM framework, benefits less from the increased number of visual features detected at reduced speed in the first experiment, yielding less accurate trajectory estimates.

Our observations from the second experiment suggest that if the robot uses a gait that protects it from strong impacts and shaky motions, then the terrain type does not influence either the ATE or RPE significantly. However, the abrupt motions during the locomotion may have profound effects on the performance. From both experiments, we can draw a conclusion that as the motion gets smoother, the results get better for all the evaluated systems, but the BA-based architectures make the best use of the good quality features.

Further research on RGB-D-based SLAM for walking robots in challenging terrain should attempt to combine the best features of the existing systems. Whereas the ORB-SLAM2 is perhaps the most efficient and universal BA-based SLAM available now as open-source, it should benefit from the integration of such solutions as the fast visual odometry replacing the simple sensor motion model. Also better handling of the depth measurements uncertainty could allow this system to take more advantage of the RGB-D sensing. Another promising research direction for SLAM on walking robots is the integration of SLAM from RGB-D data and the IMU measurements. As demonstrated in Fischer *et al.* (2016) for the stereo-based S-PTAM system, even a loosely coupled approach to IMU data integration improves the localization accuracy of the walking robot, helping to overcome problems due to abrupt motions. Very recently, the monocular ORB-SLAM has been extended by tightly coupling an IMU and pre-integrating the inertial measurements directly in the BA (Mur-Artal and Tardos, 2017). However, such approach still´ has to be implemented for the RGB-D-based SLAM on a walking robot.

Acknowledgements

This research was funded by the National Science Centre, Poland under grant 2013/09/B/ST7/01583. The work of Michał Nowicki was funded by the National Science Centre, Poland under grant 2015/17/N/ST6/01228. M. Nowicki received also PhD scholarship 2016/20/T/ST7/00396 from the National Science Centre, Poland. The work of Jan Faigl and Petr Cˇ´ızek was supported by theˇ Czech Science Foundation (GACR) under research project No. 15-09600Y.ˇ

References

Belter, D. and Skrzypczynski, P. (2013), "Precise self-localization of a walking robot on rough´ terrain using parallel tracking and mapping", *Industrial Robot: An International Journal*, Vol. 40 No. 3, pp. 229–237.

Belter, D. and Walas, K. (2014), "A compact walking robot – flexible research and development platform", in Szewczyk, R. *et al.* (Eds.) *Recent Advances in Automation, Robotics and Measuring Techniques*, AISC 267, Springer, Zurich, pp. 343–352.¨

Belter, D., Nowicki, M. and Skrzypczynski, P. (2015a), "On the performance of pose-based´ RGB-D visual navigation systems", in Cremers, D. *et al.* (Eds.) *Computer Vision – ACCV 2014*, LNCS 9004, Springer, Zurich, pp. 407–423.¨

Belter, D., Nowicki, M., Skrzypczynski, P., Walas, K., and Wietrzykowski, J. (2015b),´ "Lightweight RGB-D SLAM system for search and rescue robots", in Szewczyk, R. *et al.* (Eds.) *Progress in Automation, Robotics and Measuring Techniques*, AISC 351, Springer, Zurich, pp. 11–21.¨

Belter D., Nowicki M. and Skrzypczynski P. (2016a), “Improving accuracy of feature-based´ RGB-D SLAM by modeling spatial uncertainty of point features”, *Proc. IEEE Int. Conf. on Robotics and Automation*, Stockholm, Sweden, pp. 1279–1284.

Belter D., Labe¸cki P. and Skrzypczynski P. (2016b), “Adaptive motion planning for autonomous´ rough terrain traversal with a walking robot”, *Journal of Field Robotics*, Vol. 33 No. 3, pp. 337–370.

Belter D., Kostusiak A., Nowicki M. and Skrzypczynski, P. (2016c), “Real-time SLAM from´ RGB-D data on a legged robot: An experimental study”, in Tokhi, M. O. and Virk, G. S. (Eds.) *Advances in Cooperative Robotics*, World-Scientific, Singapore, pp. 320–328.

Cˇ´ızek, P. and Faigl, J. (2016), “On localization and mapping with RGB-D sensor and hexapodˇ walking robot in rough terrains”, *Proc. IEEE Int. Conf. on Systems, Man, and Cybernetics*, Budapest, Hungary, pp. 2273–2278.

Dryanovski, I., Valenti, R. and Xiao, J. (2013), “Fast visual odometry and mapping from RGB-D data”, *Proc. IEEE Int. Conf. on Robotics and Automation*, Karlsruhe, Germany, pp. 5704– 5711.

Durrant-Whyte, H. F. and Bailey, T. (2006), “Simultaneous localization and mapping (Part II)”, *IEEE Robotics & Automation Magazine*, Vol. 13 No. 3, pp. 99-108.

Endres, F., Hess, J., Sturm, J., Cremers, D. and Burgard W. (2014), “3-D Mapping with an RGB-D Camera”, *IEEE Transactions on Robotics*, Vol. 30 No. 1, pp. 177–187.

Fischer, T., Pire, T., Cˇ´ızek, P., De Cristˇ oforis, P. and Faigl, J. (2016), “Stereo vision-based local-´ ization for hexapod walking robots operating in rough terrains”, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Daejeon, Korea, pp. 2492–2497.

Grisetti, G., Kummerle, R., Stachniss, C. and Burgard, W. (2010), “A tutorial on graph-based¨ SLAM”, *IEEE Intelligent Transportation Systems Magazine*, Vol. 2 No. 4, pp. 31–43.

Kerl, C., Sturm, J. and Cremers, D. (2013), “Dense visual SLAM for RGB-D cameras”, *proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, pp. 2100–2106.

Klein, G. and Murray, D. (2007), “Parallel tracking and mapping for small AR workspaces”, *Proc. Int. Symp. on Mixed and Augmented Reality*, Nara, Japan, pp. 225–234.

Krajnik, T., Nitsche, M., Faigl, J., Vanek, P., Saska, M., Preucil, L., Duckett, T. and Mejail, M. (2014), “A practical multirobot localization system”, *Journal of Intelligent and Robotic Systems*, Vol. 76 No. 3-4. pp. 539–562.

Kummerle, R., Grisetti, G., Strasdat, H., Konolige, K. and Burgard, W. (2011), “g¨ ^{2}o: A general framework for graph optimisation”, *Proc. IEEE Int. Conf. on Robotics and Automation*, Shanghai, China, pp. 3607–3613.

Mrva, J. and Faigl, J. (2015), “Tactile sensing with servo drives feedback only for blind hexapod walking robot”, *Proc. 10th Int. Workshop on Robot Motion and Control (RoMoCo)*, Poznan,´ pp. 240–245.

Mur-Artal, R., Montiel, J. M. M. and Tardos, J. D. (2015), “ORB-SLAM: a versatile and accurate´ monocular SLAM system”, *IEEE Transactions on Robotics*, Vol. 31 No. 5, pp. 1147–1163.

Mur-Artal, R. and Tardos, J. D. (2016),´ *ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras*, arXiv preprint, arXiv:1610.06475v1.

Mur-Artal, R. and Tardos, J. D. (2017), “Visual-inertial monocular SLAM with map reuse”,´ *IEEE Robotics and Automation Letters*, Vol. 2 No. 2, pp. 796–803.

Pire, T., Fischer, T., Civera, J., De Cristoforis, P. and Berlles J. J. (2015), “Stereo parallel tracking´ and mapping for robot localization”, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Hamburg, Germany, pp. 1373–1378.

Ronnau A., Kerscher T., Ziegenmeyer M., Schamm T., Zoellner J. and Dillmann R. (2009), “Lo-¨ calisation of a six-legged walking robot in rough terrain by ToF-odometry”, *Proc. IARP Robotics for Risky Interventions and Environmental Surveillance (RISE)*, Brussels, Belgium, pp. 12–14.

Scherer, S. and Zell, A. (2013), “Efficient onboard RGBD-SLAM for autonomous MAVs”, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Tokyo, Japan, pp. 1062–1068.

Schmidt, A. and Kasinski, A. (2010), “The visual SLAM system for a hexapod robot”, in Bolc,´

L. *et al.* (Eds.) *Computer Vision and Graphics*, LNCS 6375, Springer, Zurich, pp. 260–267.¨

Schmidt, A., Kraft, M., Fularz, M. and Domagala, Z. (2013), “Comparative assessment of point feature detectors and descriptors in the context of robot navigation”, *Journal of Automation, Mobile Robotics & Intelligent Systems*, Vol. 7 No. 1, pp. 11–20.

Schmidt, A., Kasinski, A., Kraft, M., Fularz, M. and Domagala, Z. (2014), “Calibration of the multi-camera registration system for visual navigation benchmarking”, *Int. Journal of Advanced Robot. Syst.*, Vol. 11 No. 83.

Skrzypczynski, P. (2012), “Laser scan matching for self-localization of a walking robot in man-´ made environments”, *Industrial Robot: An International Journal*, Vol. 39 No. 3, pp. 242– 250.

Stelzer, A., Hirschmuller, H. and G¨ orner, M. (2012), “Stereo-vision-based navigation of a six-¨ legged walking robot in unknown rough terrain”, *Int. Journal of Robotics Research*, Vol. 31 No. 4, pp. 381–402.

Strasdat, H., Montiel, J. M. M. and Davison, A. J. (2012), “Visual SLAM: Why filter?”, *Image and Vision Computing*, Vol. 30 No. 2, pp. 65–77.

Sturm, J., Engelhard, N., Endres, F., Burgard, W. and Cremers, D. (2012), “A benchmark for the evaluation of RGB-D SLAM systems”, *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Vilamoura, Portugal, pp. 573–580.

Triggs, B., McLauchlan, P. F., Hartley, R. I. and Fitzgibbon, A. W. (2000) “Bundle adjustment – a modern synthesis”, *Vision Algorithms: Theory and Practice*, LNCS 1883, Springer, London, pp. 298–372.

Umeyama, S. (1991), “Least-squares estimation of transformation parameters between two point patterns”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13 No. 4, pp. 376–380.

Whelan T., Kaess M., Johannsson H., Fallon M., Leonard J. J. and McDonald J. (2015), “Realtime large-scale dense RGB-D SLAM with volumetric fusion”, *Int. Journal of Robotics Research*, Vol. 34 No. 4–5, pp. 598–626.

Author biographies

Michał Nowicki graduated from Poznan University of Technology in 2014 receiving the B.Sc. and M.Sc. in Automatic Control and Robotics in 2013 and 2014, respectively, and the BSc in Computer Science from the same University in 2014. Currently, he is a research assistant and Ph.D. student at the Institute of Control and Information Engineering. His research interests include computer vision, simultaneous localization and mapping, walking robots, and Android-based navigation.

Dominik Belter graduated from the Poznan University of Technology (2007). He received the Ph.D.degree in robotics from the same University in 2012. Since 2012 he is Assistant Professor at the Institute of Control and Information Engineering of the Poznan University of Technology. Dominik Belter is the author or co- author of over 50 technical papers in the fields of robotics and computer science. His research interests include walking robots, machine learning, and soft computing.

Aleksander Kostusiak graduated from Poznan University of Technology in 2014 receiving the B.Sc. and M.Sc. in Automatic Control and Robotics in 2013 and 2014, respectively. He is a Ph.D. student at the Institute of Control and Information Engineering. His research interests include computer vision in robotics and simultaneous localization and mapping.

Petr Čížek is a Ph.D. student at the Artificial Intelligence Center, Czech Technical University in Prague. His research interests include autonomous navigation, computer vision and sensory data fusion in intelligent mobile robotics.

Jan Faigl graduated in technical cybernetics at the Czech Technical University in Prague (CTU) in 2003. He received Ph.D. in artificial intelligence and biocybernetics in 2010. He is Associate Professor of computer science at the CTU and head of Computational Robotics Laboratory at the Artificial Intelligence Center (AIC) at the CTU. Prof. Faigl is the author or co-author of more than 100 papers in the fields of robotics, neural networks, and computer science. His current research interests include autonomous navigation, neural networks and computational intelligence methods in intelligent mobile robotics.

Piotr Skrzypczynski graduated from the Poznan University of Technology (1993). He received the Ph.D. and D.Sc. degrees in Robotics from the same University in 1997 and 2007, respectively. He is Professor at the Institute of Control and Information Engineering of the Poznan University of Technology, and head of the Mobile Robotics Lab. Prof. Skrzypczynski is the author or co-author of more than 160 technical papers in the fields of robotics and computer science. His research interests include: autonomous mobile robots, simultaneous localization and mapping, multisensor fusion, and computational intelligence methods in robotics.