# Evaluating Map-Based RGB-D SLAM on an Autonomous Walking Robot

**Dominik Belter, Michał Nowicki and Piotr Skrzypczyński**

**Abstract** This paper demonstrates an application of a Simultaneous Localization and Mapping algorithm to localize a six-legged robot using data from a compact RGB-D sensor. The algorithm employs a new concept of combining fast Visual Odometry to track the sensor motion, and a map of 3-D point features and robot poses, which is then optimized. The focus of the paper is on evaluating the presented approach on a real walking robot under supervision of a motion registration system that provides ground truth trajectories. We evaluate the accuracy of the estimated robot trajectories applying the well-established methodologies of Relative Pose Error and Absolute Trajectory Error, and investigate the causes of accuracy degradation when the RGB-D camera is carried by a walking robot. Moreover, we demonstrate that the accuracy of robot poses is sufficient for dense environment mapping in 3-D.

**Keywords** RGB-D SLAM · Factor graph · OctoMap · Walking robot

## 1 Introduction

Autonomous walking robots need accurate self-localization with respect to six degrees of freedom in order to plan and execute their complicated motion patterns in previously unknown environments. Such robots start to be considered for Urban Search And Rescue (USAR) missions, due to their high mobility, which includes the ability to overcome rough terrain and man-made obstacles. In an USAR mission the robot has to plan its motion, create a map of unknown environment, and self-localize

D. Belter (✉) · M. Nowicki · P. Skrzypczyński
Institute of Control and Information Engineering,
Poznań University of Technology, ul. Piotrowo 3A, 60-965 Poznań, Poland
e-mail: dominik.belter@put.poznan.pl

M. Nowicki
e-mail: michal.nowicki@put.poznan.pl

P. Skrzypczyński
e-mail: piotr.skrzypczynski@put.poznan.pl

R. Szewczyk et al. (eds.), *Challenges in Automation, Robotics and Measurement Techniques*, Advances in Intelligent Systems and Computing 440, DOI 10.1007/978-3-319-29357-8_42

with respect to this map. These functions need to be tightly integrated in order to achieve full autonomy and to ensure reliable motion execution [6]. Our earlier experiments demonstrated that it is necessary to achieve self-localization accuracy of about the size of the robot's foot to ensure robust registration of the range data into an elevation map that enables foothold planning [1]. Moreover, an autonomous walking robot requires a dense representation of the terrain for planning. While some state-of-the-art 3-D self-localization systems build voxel-based maps, they usually rely on massive parallel processing to handle the huge amount of range data [22]. Such approach is not suitable for most legged robots, due to the power consumption of a high-end GPU. Therefore, an autonomous legged robot needs a Simultaneous Localization and Mapping (SLAM) algorithm that is accurate, runs in real-time without hardware acceleration, and is coupled with a dense environment mapping.

In this paper we investigate a RGB-D SLAM system based on factor graph optimization implemented on an autonomous six-legged robot. This system uses a compact and affordable structured light RGB-D sensor. This class of sensors has been positively evaluated in our previous works for self-localization of USAR robots [5], and for legged robots in particular [13]. We consider a new architecture of the RGB-D SLAM system, which differs from our recent developments [3, 5] and other similar solutions [8, 9] by employing a map of 3-D point features that are included in the factor graph optimization process. The flexible structure of the factor graph enables real-time optimization of the map in a background process delegated to a separate thread [4]. While there are other RGB-D SLAM systems that use a map of point features [12, 15], they directly match the current perception to features in a large map, whereas we use a fast Visual Odometry (VO) algorithm to obtain a first guess of the camera pose, which improves matching robustness.

The evaluation focuses on the trajectory accuracy achieved in experiments on a real walking robot. We demonstrate that the proposed approach to RGB-D SLAM yields robot pose estimates that are accurate enough to build a dense, 3-D map of the environment employing a voxel-based mapping algorithm. Moreover, we identify the practical problems related to using a RGB-D camera mounted on a legged robot, and show that tighter coupling between the SLAM system and the control strategy of the robot is beneficial to the accuracy of the estimated trajectories.

## 2 Map-Based RGB-D SLAM

The Poznan University of Technology (PUT) RGB-D SLAM system is based on the factor graph optimization approach and a persistent, scalable map of 3-D point features. This system follows the non-linear optimization approach to state estimation, which is currently considered to be the state-of-the-art method in SLAM. The software[1] architecture of PUT RGB-D SLAM is presented in Fig. 1a. The front-end

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

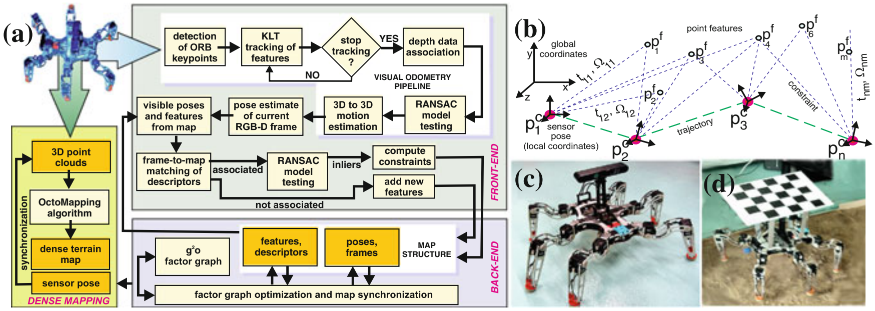

**Fig. 1** Block scheme of the PUT RGB-D SLAM with dense mapping system (**a**), structure of the feature-based map (**b**), and the Messor II robot with the Asus Xtion (**c**) and the chessboard marker (**d**)

and the back-end work asynchronously, and get synchronized only on specific events. Owing to this architecture the system efficiently uses a multi-core CPU without additional hardware acceleration.

## 2.1 Back-End

In PUT RGB-D SLAM the back-end holds a map (Fig. 1b), which contains 3-D point features augmented by their local visual descriptors [16]. The map contains also the sensor poses related to the keyframes, which were used to extract the map features. The back-end is based on the g$^2$o general graph optimization library [11]. The factor graph for optimization in the back-end is constructed from the map data. The observed features and the sensor poses are related by measurement constraints in $\mathbb{R}^3$. The initial locations of the sensor poses $\mathbf{p}^c_i$ $(i = 1 \ldots n)$ are computed using the VO pipeline. The sensor poses are represented by Cartesian positions $(x, y, z)$ and quaternions $(q_w, q_x, q_y, q_z)$ parameterizing the sensor orientation. The factor graph has two types of vertices: $\mathbf{p}^f$ representing the point features, and $\mathbf{p}^c$ representing the sensor poses. The initial positions of features $\mathbf{p}^f_j$ $(j = 1 \ldots m)$ in the sensor coordinate system are determined from the RGB-D measurements. Sensor poses are expressed in the global frame of the map, while the features are represented relatively to the poses from which they have been detected. The edge $\mathbf{t}_{ij} \in \mathbb{R}^3$ represents measurement constraints between the $i$th pose and the $j$th feature, and the edge $\mathbf{T}_{ik} \in \text{SE}(3)$ is a rigid transformation constraint resulting from the estimated motion between poses $i$ and $k$. The pose-to-pose constraints are introduced to the factor graph whenever the number of re-observed features in the map is below a threshold. The accuracy of each constraint is represented by its information matrix $\mathbf{\Omega}$, which can be obtained by inverting the covariance matrix of the measurement, or can be set to identity, if we assume anisotropic spatial uncertainty in the measurements. The sequence of the sensor (robot) poses $\mathbf{p}^c_1, \ldots, \mathbf{p}^c_n$ and feature positions $\mathbf{p}^f_1, \ldots, \mathbf{p}^f_m$ that satisfies the set of constraints is computed as:

$$\operatorname*{argmin}_{\mathbf{p}} F = \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{e}^T_{f(i,j)} \boldsymbol{\Omega}_{ij} \mathbf{e}_{f(i,j)} + \sum_{i,k=1, i \neq k}^{n} \mathbf{e}^T_{c(i,k)} \boldsymbol{\Omega}_{ik} \mathbf{e}_{c(i,k)} \tag{1}$$

where $\mathbf{e}_{f(i,j)} = e(\mathbf{p}^c_i, \mathbf{p}^f_j, \mathbf{t}_{ij})$ and $\mathbf{e}_{c(i,k)} = e(\mathbf{p}^c_i, \mathbf{p}^c_k, \mathbf{T}_{ik})$ are error functions for the feature-to-pose and pose-to-pose constraints, respectively. These functions are computed for the estimated pose of the vertex and measured pose of the vertex which comes out from the measurement $\mathbf{t} \in \mathbb{R}^3$ for feature-to-pose or $\mathbf{T} \in SE(3)$ for pose-to-pose constraints. We use the implementation of Preconditioned Conjugate Gradient method from the g$^2$o library [11] to solve (1). We buffer new sensor poses, features, and measurements in a smaller temporary graph during the graph optimization process. Thus, we can synchronize the optimized factor graph with the map of features when the back-end finishes the on-going graph optimization session.

## 2.2 Front-End

The core part of the front-end is a fast VO pipeline that yields a sensor displacement guess. The VO estimates the SE(3) transformation between two poses employing sparse optical flow tracking with the Lucas-Kanade algorithm [19]. We use ORB features [14] to initialize the keypoints for tracking, as due to the computing efficiency considerations local descriptors of the features extracted in the VO pipeline are then re-used for matching between the features from a new keyframe and the map. Correct matching features in two keyframes are estimated from the initial set of keypoints associated by optical flow tracking by using the preemptive RANSAC framework. The RANSAC samples three pairs of keypoints from the initial set, and using the Umeyama algorithm [21] estimates the candidate transformation, which is then verified using the remaining pairs of keypoints. The implementation of the tracking-based VO algorithm is described in more detail in our earlier work [3].

Once the current sensor pose is computed by VO, the front-end attempts to match features detected in the current keyframe to features in the map. To robustly determine feature-to-map correspondences the guided matching approach is applied—the feature matches are considered only in a small neighborhood of the predicted features in the image plane, and then verified by typical matching based on the ORB descriptors. With a set of possible matches, the inliers are determined again by RANSAC. Thanks to the constraints established between the current keyframe and the features already included in the map (from previous, possibly distant keyframes) our SLAM algorithm handles local metric loops implicitly, without the need to identify already visited places by their appearance. However, closing large loops by this method is problematic, as the amount of drift in the predicted sensor pose may be too large for the guided matching approach.

## 2.3 Dense Environment Mapping

The PUT RGB-D SLAM system creates a sparse map of 3-D features, which is an efficient environment representation for self-localization. However, for motion planning the autonomous walking robot needs a dense 3-D, or at least 2.5-D environment representation [6]. Therefore, we coupled our SLAM system with a 3-D occupancy grid mapping algorithm. Having in the SLAM map the optimized robot/sensor poses from which the measurements have been made, we project point clouds associated to the keyframes into a global coordinate system. Although the point cloud or surfel-based representation of the RGB-D data can provide a good visualization of the scene, they are highly redundant and inefficient whenever the map has to be queried by the motion planning algorithm for occupancy information at particular coordinates. Thus, we convert the point clouds to the efficient octree-based OctoMap occupancy map representation proposed in [10]. In OctoMap the voxels are stored in a tree structure that results in a compact memory representation and allows the planning algorithms to query the occupancy map at various resolutions. Moreover, OctoMap is a probabilistic framework, which efficiently handles noisy measurements and uncertain sensor poses. We use the algorithm and open source software from [10] added to our system to create OctoMaps from registered point clouds. In the experiments presented in this paper we have used 10 mm voxel grid resolution.

## 2.4 Implementation on the Walking Robot

The Messor II robot [2] uses the Asus Xtion Pro Live RGB-D sensor based on the PrimeSense structured light camera—the same that is employed in the more popular Kinect sensor, which is however bigger, heavier, and requires an additional power source. The compact and lightweight Asus Xtion, which is powered by USB only is much more suitable for a walking robot.

On the walking robot the Asus Xtion is mounted on a mast fabricated using the 3-D printing technology (Fig. 1c). The sensor is located at the elevation of 40 cm above the ground, assuming the neutral posture of the robot. Since the RGB-D sensor should also measure the ground area in front of the robot for terrain mapping [6], it is slightly tilted down. This configuration allows the sensor to perceive the shape of the terrain at some distance from the robot, and to observe objects located farther away. Such objects are important for SLAM, as they provide more salient features than the terrain surface. Although the robot has an Inertial Measurements Unit (IMU) that can estimate the attitude of the robot's trunk with respect to the gravity vector, in the present implementation of our SLAM system these measurements are not used.

Currently the Messor II robot has only an embedded computer (PandaBoard) for its control software, thus the SLAM system runs on a laptop PC. Although for the experiments presented in this paper the robot was tethered to the laptop and an external power source, it is possible to use a WiFi connection and on-board batteries for full autonomy.

## 3 Evaluation Procedure

There are publicly available data sets, such as the TUM RGB-D benchmark [20], which facilitate benchmarking of RGB-D SLAM algorithms and enable comparison to state-of-the-art approaches. The PUT RGB-D SLAM was evaluated on the TUM RGB-D benchmark, demonstrating on some sequences the trajectory reconstruction accuracy better than the best results published so far in the literature [4]. However, the TUM RGB-D benchmark and other similar data sets, obtained either using a handheld Kinect sensor or a sensor mounted on a wheeled robot, do not allow us to evaluate robustness of our SLAM system to problems specific to walking robots: legged locomotion is discrete, which often causes uncontrolled oscillations of the attitude and vibratory motion of the platform, which makes it hard to keep the line of sight of any on-board sensor horizontal. Moreover, the RGB-D sensor on a walking robot is usually mounted at quite low height (a tall mast cannot be used due to the robot stability issues), which limits the effective field of view.

Therefore, we decided to test the PUT RGB-D SLAM on a real six-legged robot.[2] Some results from tests on the same walking robot were already published for the earlier, pose-based variant of our RGB-D SLAM [5], however, these results were only qualitative due to the lack of ground truth trajectories. We demonstrate the trajectory reconstruction accuracy achieved on the Messor II robot with quantitative results, applying the Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) metrics proposed in [20]. The ATE error compares the distance between the estimated and ground truth trajectories, and is computed from the Root Mean Square Error (RMSE) for all nodes of the ground truth and the recovered trajectory. The RPE error corresponds to the local drift of the trajectory. To compute RPE the relative transformation between the neighboring point of the ground truth trajectory $\mathbf{T}_i^{\mathrm{GT}}$ and the recovered trajectory $\mathbf{T}_i$ is computed, and the relative error for each point of the trajectory is given by:

$$\mathbf{E}_i = \left(\mathbf{T}_i^{\mathrm{GT}}\right)^{-1} \mathbf{T}_i. \tag{2}$$

Taking the translational or rotational part of $\mathbf{E}_i$ we obtain the translational or rotational RPE, respectively.

The ground truth trajectories were obtained using an overhead system of cameras [17]. Although this multi-camera vision system consists of five high-resolution Basler acA1600 cameras equipped with low-distortion, aspherical 3.5 mm lenses, for the presented experiments we have used only the central camera, which was possible because of the small area of the terrain mockup traversed by the walking robot (Fig. 2a). Employing only one camera simplified the registration procedure, as mutual calibration of the overhead cameras was not necessary [18]. The central camera was calibrated using a chessboard pattern according to the rational lens model [7]. We synchronized the acquisition of RGB-D frames by the Asus Xtion on the robot with the frame rate of the overhead camera (they both work at 15 Hz, due to

[2] Data set is publicly available at http://lrm.put.poznan.pl/putslam/.

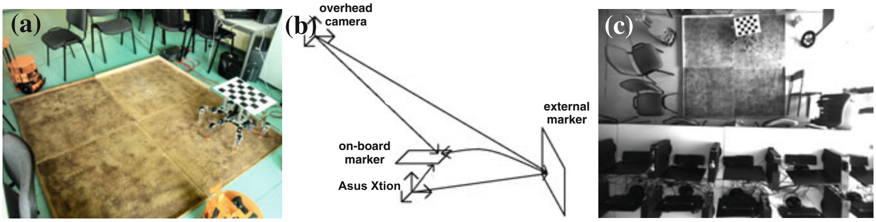

**Fig. 2** Evaluation experiment: experimental set up with a flat terrain mockup and the Messor II robot (**a**), system calibration concept (**b**), and an exemplary view from the overhead camera of the PUT Ground Truth system (**c**)

limitations of the camera) to avoid errors due to trajectory interpolation, which arise in other RGB-D data sets [20].

The Messor II robot used in the experiments was equipped with the Asus Xtion sensor and a rigidly attached chessboard marker facing up (Fig. 1d). The rigid transformation between the Asus Xtion coordinate system, and the coordinates of the chessboard marker was estimated applying the calibration algorithm from [18], which uses images of an external calibration marker observed by the on-board sensor and images registered by an external camera observing both the external marker, and the marker attached to the robot (Fig. 2b). During the experiments overhead camera images (Fig. 2c) were registered synchronously with the Asus Xtion frames. The chessboard pattern of known size was extracted from these images and its pose with respect to the overhead camera was determined. Then, this pose was transformed to the pose of the sensor using the transformations obtained by calibration. All artificial markers were used only by the PUT Ground Truth system, and the operation of PUT RGB-D SLAM didn't depend on them.

## 4 Results

All experiments with the Messor II robot have been conducted on a flat terrain mockup of the size 2 × 2 m. Some common objects (furniture, lab equipment) have been located outside the mockup to make the environment richer in salient features, which are needed by our SLAM to operate reliably. All test trajectories resembled rectangles of the lap size similar to the width of the mockup, but executed at different translational speeds of the robot, making also more or less sharp turns at the corners of the trajectory. The layout of objects around the mockup area was different for each experiment.

In the first experiment the Messor II robot used the default tripod gait. This is the fastest statically stable gait for a six-legged machine, thus the robot moved at 95 % of its maximal speed. As the robot was teleoperated using a joystick, the step length was variable and the effective velocity of the robot's trunk with the sensor

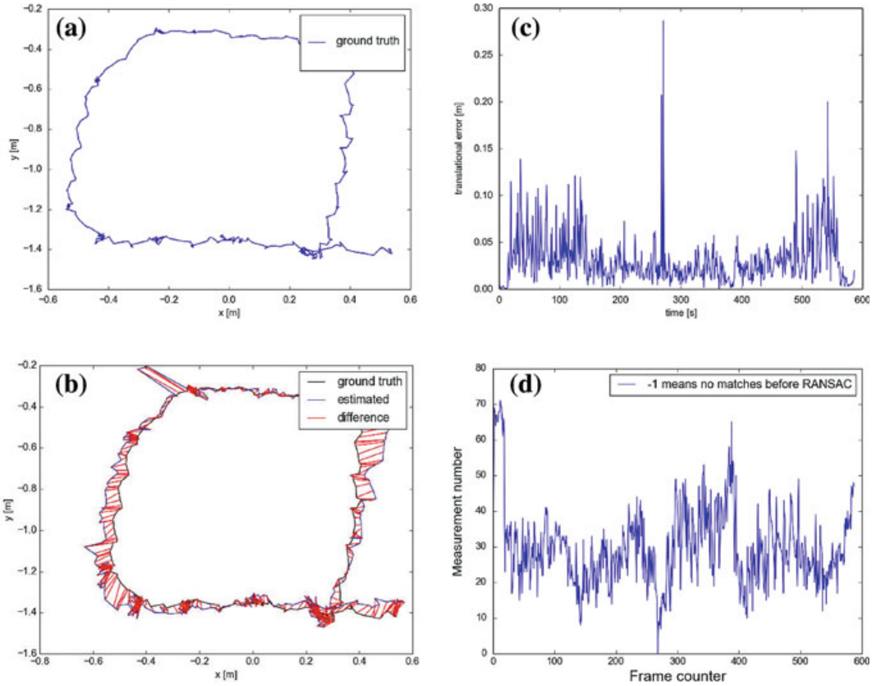

**Fig. 3** Results for the robot walking at 95 % of its maximal speed (`messor2_1`): ground truth trajectory (**a**), estimated trajectory with ATE (**b**), transitional RPE (**c**), and the number of matched map features (**d**)

varied as well, but the average velocity was about 0.15 m/s. The resulting ground truth trajectory is shown in Fig. 3a. The estimated trajectory and the ATE values are visualized in Fig. 3b. The ATE RMSE in this experiment was about 9.5 cm, which is a value bigger than expected for reliable dense mapping and motion planning. Plot of the translational RPE (Fig. 3c) reveals that at some frames (e.g. frame 280, pointed by the arrow) the relative error peaked to more than 0.25 m. The moments of increased relative error coincide with sudden drops of the number of measurements between the new frame and the map (i.e. the number of established feature-to-pose constraints, Fig. 3d), which usually means that the VO-based guess of the camera pose was very poor. This leads to sensor poses that are weakly connected to the map features, and to an increased error in the optimized trajectory, as seen at the ATE plot (Fig. 3a, pointed by the arrow).

Careful inspection of the recorded RGB frames revealed that the moments of much degraded performance directly coincide with the frames containing an excessive amount of motion blur. An example of such corrupted RGB-D data from this experiment is shown in Fig. 4. While frames (a) and (e) are sharp enough, the frame (c) between them is blurred to the extent that makes it impossible to track the keypoints with the Lucas-Kanade algorithm. The motion blur is apparently caused by

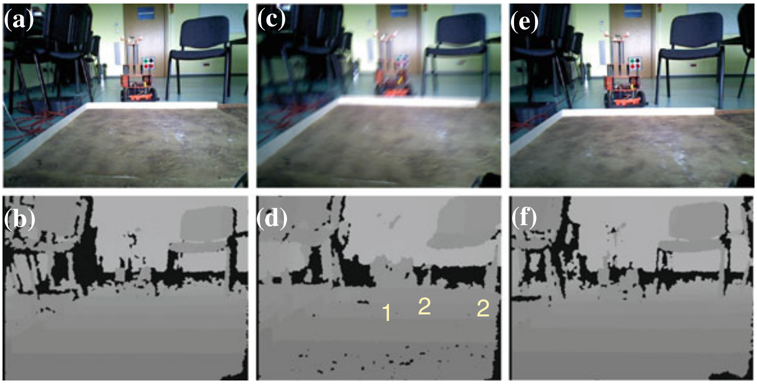

**Fig. 4** RGB (**a**, **c**, **e**) and depth (**b**, **d**, **f**) frames from the Asus Xtion mounted on the Messor II robot acquired during the experiment at 95 % of maximal speed

quite large amplitude vibrations of the robot's trunk observed when it is supported by three legs in the swing phase of the fast tripod gait. What is interesting, the motion blur degrades also the depth images, as seen in Fig. 4d—some details seen on the neighboring depth frames, like the wheeled robot (denoted '1') or legs of the chair (denoted '2') are completely blurred or missing entirely. This effect makes extraction of the 3-D point features from the blurred frames extremely unreliable.

If the speed of the robot is reduced to 45 % of the maximal value (average velocity of about 0.09 m/s) the trajectory is recovered with slightly better accuracy, as shown in Fig. 5. The robot still takes some frames when it is supported by three legs only, but due to the reduced speed, the trunk vibrations have smaller amplitude. As the result, the number of measurements between the current keyframe and the map never drops to zero (Fig. 5d).

Better accuracy and a much smoother recovered trajectory can be obtained if we couple the acquisition of RGB-D frames for SLAM with the gait of the robot. This can be fully implemented only if we let the motion execution algorithm to control the RGB-D sensor, which was impossible in the presented experiments. However, when we have changed the default tripod gait to a slower crawl gait, which moves only one leg of the robot at once, we were able to avoid excessive motion blur. In this case the average velocity of the robot was only 0.05 m/s, but we were able to achieve the ATE RMSE of about 5 cm, which seems to be enough for terrain mapping (almost the size of the robot's foot). Results of this experiment are depicted in Fig. 6. Table 1 summarizes the quantitative results for all the three experiments presented in this paper.

In order to demonstrate that the self-localization accuracy achieved by PUT RGB-D SLAM on the real walking robot is sufficient for dense environment mapping we produced an voxel-based occupancy grid from the points measured in the third experiment (`messor2_3` trajectory). In Fig. 7a and b we show screenshots from a dynamic visualization of the point cloud registered with the recovered trajectory.

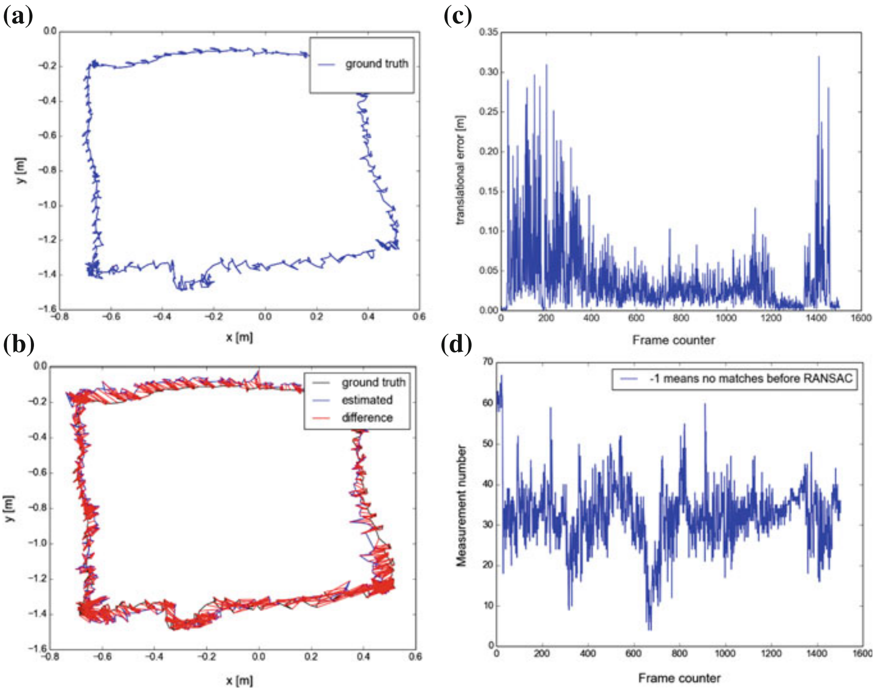

**Fig. 5** Results for the robot walking at 45 % of its maximal speed (`messor2_2`): ground truth trajectory (**a**), estimated trajectory with ATE (**b**), transitional RPE (**c**), and the number of matched map features (**d**)

This visualization can run on-line along with PUT RGB-D SLAM, and is presented in a movie clip available at http://lrm.cie.put.poznan.pl/aut2016.mp4. The registered point clouds are transformed into the more compact OctoMap representation presented in Fig. 7c and d using the OctoMap viewer [10].

## 5 Conclusions

A new approach to the RGB-D SLAM problem, which employs the factor graph optimization and builds a persistent map of 3-D features has been presented and evaluated for the use on a real autonomous walking robot. The SLAM algorithm uses visual odometry to compute the sensor motion guess, instead of direct tracking of the sensor against the map. The VO pipeline implemented using fast optical flow tracker results in a very accurate RGB-D SLAM running at the RGB-D camera frame rate without GPU acceleration, which makes it suitable for on-board implementation in various robots that require accurate self-localization in 3-D, such as autonomous walking robots in USAR missions.

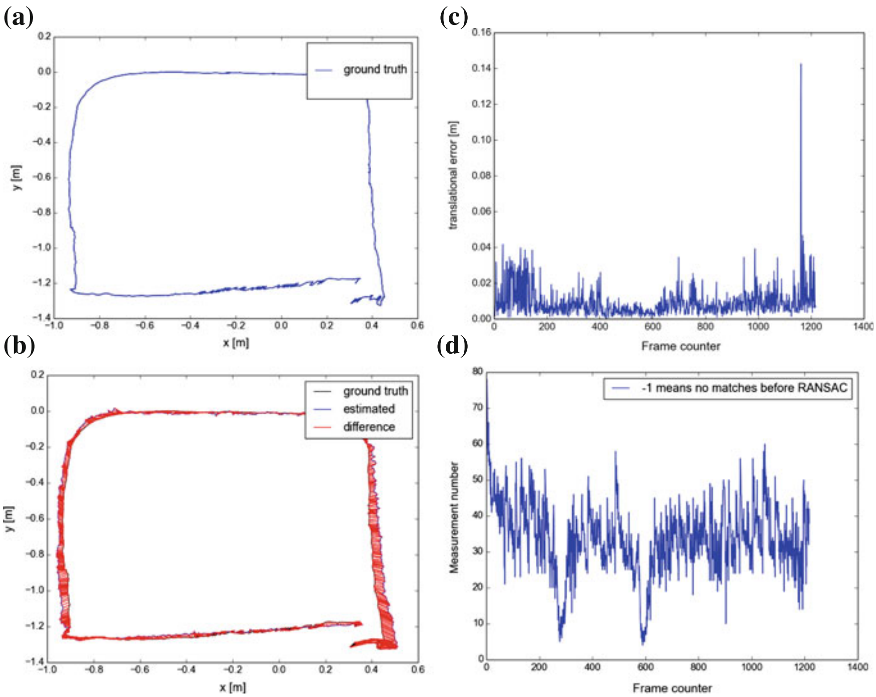

**Fig. 6** Results for the robot acquiring RGB-D frames when it was in the support phase of crawl gait (`messor2_3`): ground truth trajectory (**a**), estimated trajectory with ATE (**b**), transitional RPE (**c**), and the number of matched map features (**d**)

**Table 1** ATE and RPE for the PUT RGB-D SLAM measured on three sequences

| Messor II experiment RGB-D data sequence | ATE RMSE (m) | ATE std dev. (m) | Translat. RPE RMSE (m) | Rotat. RPE RMSE (deg) |
|---|---|---|---|---|
| `messor2_1` | 0.095 | 0.052 | 0.054 | 4.96 |
| `messor2_2` | 0.069 | 0.033 | 0.041 | 5.66 |
| `messor2_3` | 0.053 | 0.025 | 0.012 | 5.42 |

Our experiments revealed that the most pronounced problems specific to the multi-legged platform are the unstable motion and excessive vibrations, which corrupt the RGB-D data acquired in-motion. One solution to these problems, that was to some extent implemented in our experiments is to couple the frame acquisition process with the gait control of the robot. Another one, which we plan to research in the near future is to compensate the motion of the platform by using data from the on-board IMU. Obviously, both approaches may be then combined for even better results.

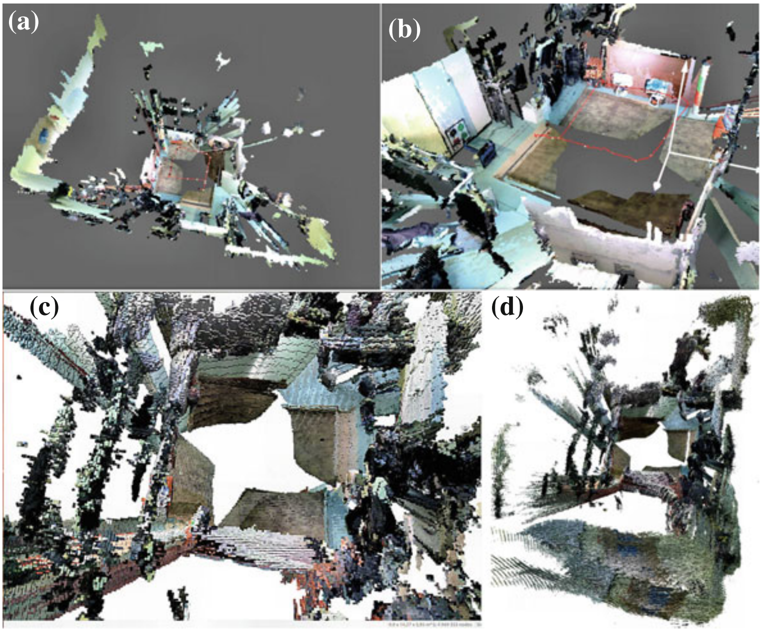

**Fig. 7** Visualization of RGB-colored point clouds registered with the estimated robot trajectory `messor2_3` (**a**, **b**), and the RGB-colored voxel-based map created from these data with the OctoMapping algorithm (**c**, **d**)

We have also demonstrated experimentally, that the accuracy of PUT RGB-D SLAM is good enough for dense mapping of the environment from a real walking robot. In further research we will demonstrate that the obtained occupancy map can be used for efficient motion planning with the algorithms that have been already tested on simpler elevation grid maps [6].

**Acknowledgments** This work was financed by the Polish National Science Centre under decision DEC-2013/09/B/ST7/01583.

## References

1. Belter, D., Skrzypczyński, P.: Precise self-localization of a walking robot on rough terrain using parallel tracking and mapping. Ind. Robot: Int. J. **40**(3), 229–237 (2013)
2. Belter, D., Walas, K.: 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, pp. 343–352. Springer (2014)
3. Belter, D., Nowicki, M., Skrzypczyński, P.: On the performance of pose-based RGB-D visual navigation systems. In: Cremers, D., et al. (eds.) Computer Vision—ACCV 2014, LNCS 9004, pp. 407–423. Springer (2015)

4. Belter, D., Nowicki, M., Skrzypczyński, P.: Accurate map-based RGB-D SLAM for mobile robots. In: Reis, L., et al. (eds.) Robot 2015: Advances in Robotics, AISC. Springer (2015)
5. Belter, D., Nowicki, M., Skrzypczyński, P., Walas, K., Wietrzykowski, J.: 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, pp. 11–21. Springer (2015)
6. Belter, D., Labęcki, P., Skrzypczyński, P.: Adaptive motion planning for autonomous rough terrain traversal with a walking robot. J. Field Robot., Wiley Early View (2015)
7. Claus, D., Fitzgibbon, A.: A rational function lens distortion model for general cameras.In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (2005)
8. Endres, F., Hess, J., Sturm, J., Cremers, D., Burgard, W.: 3-D mapping with an RGB-D camera. IEEE Trans. Robot. **30**(1), 177–187 (2014)
9. Henry, P., Krainin, M., Herbst, E., Ren, X., Fox, D.: RGB-D mapping: using kinect-style depth cameras for dense 3D modeling of indoor environments. Int. J. Robot. Res. **31**(5), 647–663 (2012)
10. Hornung, A., Wurm, K., Bennewitz, M., Stachniss, C., Burgard, W.: OctoMap: an efficient probabilistic 3D mapping framework based on octrees. Auton. Robots **34**(3), 189–206 (2013)
11. Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: g2o: a general framework for graph optimization. In: IEEE International Conference on Robotics and Automation, Shanghai, pp. 3607–3613 (2011)
12. Maier, R., Sturm, J., Cremers, D.: Submap-based bundle adjustment for 3D reconstruction from RGB-D data. In: Pattern Recognition, LNCS 8753, pp. 54–65. Springer (2014)
13. Nowicki, M., Skrzypczyński, P.: Experimental verification of a walking robot self-localization system with the Kinect sensor. J. Autom., Mobile Robot. Intell. Syst. **7**(4), 42–51 (2013)
14. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: ORB: an efficient alternative to SIFT or SURF. In: IEEE International Conference on Computer Vision, pp. 2564–2571 (2011)
15. Scherer, S., Zell, A.: Efficient onbard RGBD-SLAM for autonomous MAVs. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, pp. 1062–1068 (2013)
16. Schmidt, A., Kraft, M., Fularz, M., Domagala, Z.: Comparative assessment of point feature detectors and descriptors in the context of robot navigation. J. Autom., Mobile Robot. Intell. Syst. **7**(1), 11–20 (2013)
17. Schmidt, A., Kraft, M., Fularz, M., Domagala, Z.: The registration system for the evaluation of indoor visual SLAM and odometry algorithms. J. Autom., Mobile Robot. Intell. Syst. **7**(2), 46–51 (2013)
18. Schmidt, A., Kasinski, A., Kraft, M., Fularz, M., Domagala, Z.: Calibration of the multi-camera registration system for visual navigation benchmarking. Int. J. Adv. Robot. Syst. **11**(83) (2014)
19. Shi, J., Tomasi, C.: Good features to track. In: IEEE Conference on Computer Vision and Pattern Recognition, Seattle, pp. 593–600 (1994)
20. Sturm, J., Engelhard, N., Endres, F., Burgard, W., Cremers, D.: A benchmark for the evaluation of RGB-D SLAM systems. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, pp. 573–580 (2012)
21. Umeyama, S.: Least-squares estimation of transformation parameters between two point patterns. IEEE Trans. Pattern Anal. Mach. Intell. **13**(4), 376–380 (1991)
22. Whelan, T., Johannsson, H., Kaess, M., Leonard, J., McDonald, J.: Robust real-time visual odometry for dense RGB-D mapping. In: IEEE International Conference on Robotics and Automation, Karlsruhe, pp. 5704–5711 (2013)