Czech
Technical
University
in Prague

Faculty of Electrical Engineering
Department of Measurement

Bachelor's thesis

Autonomous Localization of Hexapod Walking Robot

Jan Bayer

June 2017

Thesis supervisor: doc. Ing. Jan Faigl, PhD.

CZECH TECHNICAL UNIVERSITY IN PRAGUE

Faculty of Electrical Engineering
Department of Measurement
Technická 2, 166 27 Prague 6, Czech Republic

Academic year **2016/2017**

BACHELOR PROJECT ASSIGNMENT

Student: **Jan Bayer**

Study programme: **Cybernetics and Robotics**
Specialisation: **Sensors and Instrumentation**

Title of Bachelor Project: **Autonomous Localization of Hexapod Walking Robot**

Autonomní lokalizace šestinohého kráčejícího robotu
(in Czech)

Guidelines:

1. Familiarize yourself with the methods of Simultaneous Localization and Mapping (SLAM) for localization of mobile robots [1, 2, 3] and existing evaluation methodologies [4, 5].
2. Propose an evaluation methodology for measuring precision of SLAM methods with hexapod walking robot operating on flat surfaces and in rough terrains.
3. Implement and experimentally evaluate selected SLAM methods, e.g., ORB-SLAM, RGB-D SLAM, S-PTAM, using RGB-D, monocular and stereo vision sensors.
4. Propose improvement of the localization precision based on active perception and active localization approaches.

Bibliography/Sources:

[1] R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System." IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, 2015.

[2] F. Endres, J. Hess, J. Sturm, D. Cremers and W. Burgard, "3D Mapping with an RGB-D Camera", IEEE Transactions on Robotics, vol. 30, no. 1, pp. 177-187, 2014.

[3] Taihú Pire, Thomas Fischer, Javier Civera, Pablo De Cristóforis and Julio Jacobo Berlles, "Stereo Parallel Tracking and Mapping for Robot Localization", Proc. of IEEE International Conference on Intelligent Robots and Systems, pp. 1373-1378, 2015.

[4] D. Belter, M. Nowicki and P. Skrzypczynski, "Evaluating Map-based RGB-D SLAM on an Autonomous Walking Robot", Progress in Automation, Robotics and Measuring Techniques, AISC, Springer, 2016.

[5] J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems", Proc. of IEEE International Conference on Intelligent Robots and Systems, pp. 573-580, 201.

Bachelor Supervisor: doc. Ing. Jan Faigl, Ph.D. (K 13136)

Valid until: September 30, 2018

L. S.

Prof. Ing. Jan Holub, Ph.D.
Head of Department

Prof. Ing. Pavel Ripka, CSc.
Dean

Prague, January 10, 2017

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 25. 5. 2017

Acknowledgement

I would like to express thanks to my thesis supervisor, doc. Ing. Jan Faigl, PhD and to Ing. Petr Čížek. They taught me a lot and provide me support for development of such a challenging project.

Abstrakt

Práce se zabývá vizuální lokalizací šestinohého kráčejícího robotu. V posledních letech bylo vyvinuto několik metod vizuální lokalizace založených na simultánní lokalizaci a mapování využívající významné body v prostředí. Některé z těchto metod využívající různé senzory se ukázaly velice efektivní pro lokalizaci kolových robotů. V této práci se zabýváme složitější lokalizací šestinohého kráčejícího robotu, kde vlivem kráčení dochází k oscilacím a rychlým rotačním pohybům, které nepříznivě ovlivňují měření senzory robotu. V práci jsou porovnány tři různé metody simultánní lokalizace a mapování využívající tři různé typy senzorů. Na základě provedených experimentů byla změřena přesnost lokalizace, kterou jsou testované metody schopny poskytnout. Provedené experimenty spočívaly v lokalizaci šestinohého kráčejícího robotu v kancelářském prostředí. Na základě ohodnocení přesnosti lokalizace můžeme konstatovat, že metody simultánní lokalizace a mapování mohou poskytovat relativně přesnou lokalizaci, zejména v kratším časovém horizontu. Pro náročnější případy navrhujeme řešení na základě aktivní lokalizace.

Klíčová slova:

SLAM, šestinohý kráčející robot, visuální odometrie, S-PTAM, ORB-SLAM, RGB-D SLAM

Abstract

Vision-based localization of hexapod walking robot is addressed in this thesis. In particular, localization methods called Simultaneous localization and mapping that are based on visual approaches have been studied. In recent years, several methods of simultaneous localization and mapping using detection of salient points in the robot's environment by various types of cameras were developed. Some of these methods were proved to be very effective for localization of wheeled robots. However, in this thesis, we address more challenging visual localization of hexapod walking robot where sensors suffer from motion blur, the robot's oscillations and fast rotations around several axes, induced by the robot's locomotion. Three different simultaneous localization and mapping methods (S-PTAM, ORB-SLAM2 and RGB-D SLAM v2) in combination with three different types of sensors are evaluated in this thesis. Moreover, we propose to evaluate performance of the localization methods by evaluation proposed in literature that has been further improved according to performed experimental deployments of the robot. Experiments with localization systems were performed on a low-cost hexapod platform in an office environment. Based on the presented results, we can conclude that the evaluated vision based localization can provide relatively accurate position estimation of the hexapod robot especially for short term localization. For more demanding cases, we propose to use enhance the position estimation by active localization approaches.

Keywords:

SLAM, hexapod robot, visual odometry, S-PTAM, ORB-SLAM, RGB-D SLAM

Contents

List of Figures

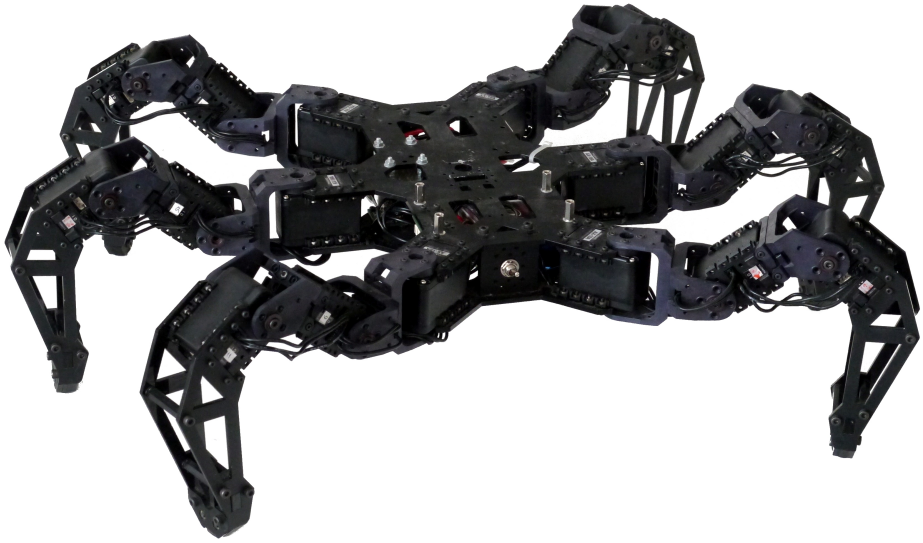
List of Tables

Part 1
Introduction

Reliable localization is an essential prerequisite for an autonomous behavior of mobile robots including unmanned aerial vehicles, wheeled robots, and also legged robots. In recent years, legged robots became popular due to their ability to traverse difficult, uneven terrain with obstacles comparable to their dimensions while interacting with the environment only by foots. This capability makes them a good choice for exploration of collapsed buildings or caves. In such cases, it is necessary to build a map of robot's surroundings by onboard sensors, which enforces utilization of relatively precise localization in comparison with the robot's dimensions.

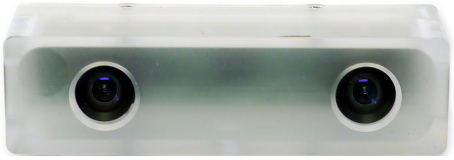
Simultaneous localization and mapping (SLAM) [1] is an approach in mobile robotics that is capable of reliable and precise localization of mobile robots. Besides, it may also provides a map of the robot's surroundings, which is beneficial especially for motion planning. The other great advantage of SLAM is ability to use only sensors mounted to the robot which results in autonomous localization without a need of any external external sensor systems such as Global Navigation Satellite System (GNSS). Although, the general problem of Simultaneous localization and mapping can be considered as theoretically solved by existing frameworks [1], there are several difficulties based on sensor's precision and computational demands, which complicate the implementation of SLAM in real on-board scenarios. Several SLAM methods were recently developed, e.g., ORB-SLAM [2], RGB-D SLAM [3], and S-PTAM [4]. These methods were proved to be effective for localization of wheeled robots. In this thesis, we address challenges of visual localization arising in deployment of these methods on a hexapod walking robot.

Several experiments in both flat and uneven terrains with the localization of hexapod walking robot equipped with multiple sensors (shown in Figure 1) were performed. As the localization, all three above metioned methods (ORB-SLAM, RGB-D SLAM or S-PTAM) were experimentaly evaluated using standard metrics utilized in SLAM evaluation [5] enhanced with minor improvements based on a SLAM behavior. The presented evaluation results of SLAM systems provide quantitative results for the comparison of SLAM systems performance in standardize and unifying way. Moreover, the evaluation yields to a set of the proposed improvement for individual methods taking into consideration an active localization approach, in which special actions to improve the localization performance are taking by considering properties and evolution of the localization precision during the robot autonomous mission.



(a) Hexapod walking robot

(b) RGB-D camera Asus Xtion



(c) Stereo camera Tara

Figure 1: Equipment used in the experimental evaluation of SLAM systems

Part 2
Vision based localization

The vision based localization is a process of acquiring the robot pose in an environment using a camera sensor. In general, localization methods can be divided into two groups: 1) absolute localization; and 2) incremental localization.

The absolute localization provides an estimate of the robot's pose independently on the previous robot's states. Examples of absolute localization systems are GPS or geodetic total station. Vision based absolute localization usually rely on recognition of the special pattern attached to the robot, e.g., April Tags [6], see Figure 2.

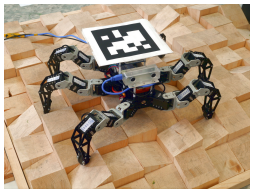


Figure 2: April Tag attached to the hexapod robot.

The incremental localization (also called dead-reckoning) has prior knowledge of the robot's starting pose. Then, each next pose is calculated from the previous pose and the robot's position change. One of the common used incremental localization methods is the odometry, which localizes a robot using its start position and counting the number of rotations of the robot's wheels utilized for the robot motion. The incremental localization can also be use to localize a hexapod walking robot using kinematic model of the robot, further supported by the Inertial measurement unit (IMU) and joint angle sensors embedded the robot's servos [7]. However, the main disadvantage of this approach is that it relies on a precise kinematic model. Due to imperfections in the model and robot's mechanical construction, the absolute error of estimated pose increases rapidly with the traveled distance. Therefore, it is more beneficial to use visual odometry, which is described in the next section.

2.1 Visual odometry

Visual odometry uses a camera mounted to the robot while processing consecutive image frames to estimate a change of the camera's pose. Each change of the camera's pose between pairs of consecutive frames is estimated by detecting correspondences between these frames. There are several methods of detecting correspondencies in images, and one of the most popular are those based on detection of salient points in the image.

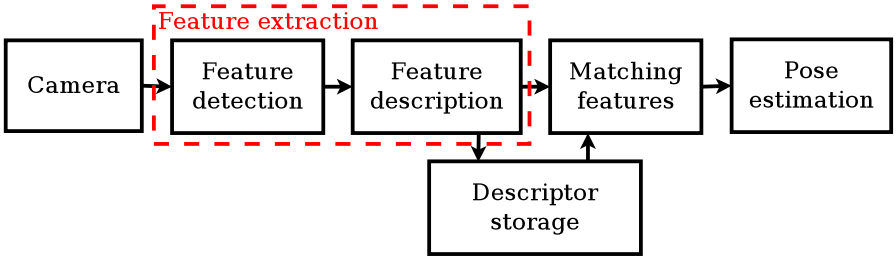


Figure 3: Overview of simple visual odometry

The salient points can be, e.g., corners, places with significant change of contrast or color.

Processing blocks of a typical feature based visual odometry are depicted in Figure 3 and it operates as follows. At first, image features are extracted, by a process that consists of two phases: feature detection and feature description. During the feature detection, limited number of image features is detected, then a local representation of each detected features (i.e., feature description) is calculated. After that, descriptors of features detected in the current frame and feature descriptors obtained from the previous frame are compared to find the same salient points in both images. The process of comparison of feature descriptors to find out, if they represent the same salient object in the environment is called feature matching. Corresponding features in both frames determine the geometrical transformation between these frames. The resulting camera pose is computed from the beginning pose and product of all following transformations calculated from the beginning frame to the current frame.

The main disadvantage of the visual odometry is that the error of the calculated transformations accumulates along the whole path of the robot. Therefore, visual odometry is not suitable for a long-term localization. However, methods based on visual odometry called Simultaneous localization and mapping (SLAM) employ more complex strategies of searching corresponding frames to allow more reliable and more accurate localization than a simple visual odometry [1]. One of the fundamental techniques is loop closure which enables to improve localization by observing a scene that was already seen.

2.2 Image features

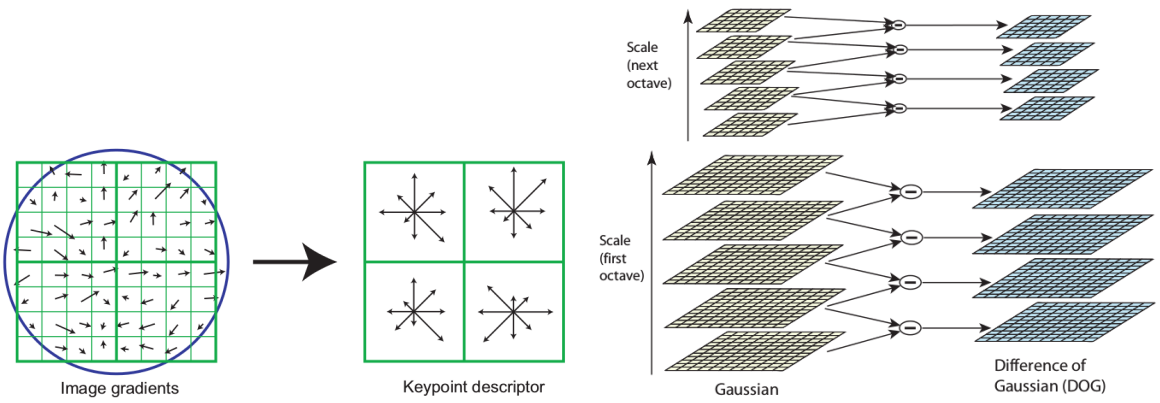
The important part of the image processing pipeline in the vision-based localization systems, e.g., such as shown in Figure 3, substantilally affects the feature based localization, is the particular type of image features [8] used to find correspondencies between the camera images. One of the most important parameters of image features is their distinctiveness, i.e., how the previously detected features can be reliably recognized when they are observed again. An additional important attribute of an image feature is the reliability to be recognized under different viewing conditions, e.g., from different viewpoints or under different illumination conditions. The invariance to changes of the

viewpoint is especially important for the loop closing. Computational complexity of image features is also important because, e.g., features that can be detected and matched faster enables the whole system to operate at a higher framerate, and thus be capable to capture fast robot motion. Four widely used features in vision-based localizations: SIFT, SURF, BRIEF, and ORB are further described in the next section to highlight their main properties.

2.2.1 SIFT features

A feature detection of the so called Scale Invariant Feature Transform (SIFT) [9] relies on a cascade filtering of the image by Gaussian function, which gradually refines features positions. Coordinates of the image features are obtained in two steps. At first, the convolution of the given input image and the substraction of two Gaussian functions is computed. Then, feature coordinates are detected as extremas of the convolution's product by Hessian matrix.

The SIFT descriptor (shown in Figure 4) is obtained from a selected square part of the image in the neighborhood of the detected feature point, where intensity gradients for each pixel are calculated. Selected part of the image is divided into 4×4 subregions, where each subregion is represented by a set of gradients with eight bins. Each contribution from the subregion is weighted by a Gaussian function to improve rotation invariance of the descriptor. Thus, each subregion can be represented by eight values, the whole SIFT feature descriptor is an 128-dimensional vector.



(a) SIFT descriptor (b) Cascade of Gaussian functions

Figure 4: SIFT descriptor construction (retrieved from [9])

SIFT features are invariant to the image scale, rotation, and change of the viewpoint. Another advantage of SIFT features is that they are highly distinctive due to their large descriptors. On the other hand, the main disadvantages of SIFT features are computational demands of the gradient calculation and also demanding feature matching which uses Euclidean distance between features.

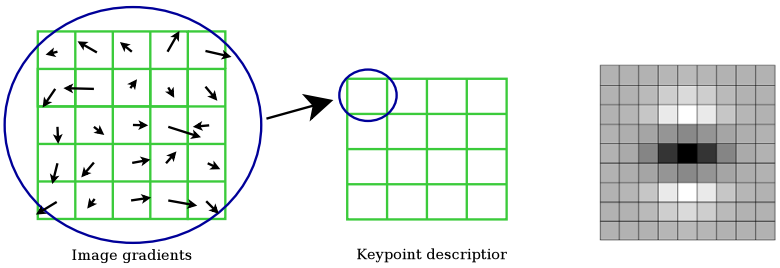
It is possible to decrease the memory consumption and computational demands of SIFT using, e.g., shorter descriptors or it is possible to compute hashes from descriptors to transform them into binary vectors. Binary vectors can be compared using the Hamming distance, which can be calculated faster than the Euclidean distance.

2.2.2 SURF

Speeded Up Robust Features (SURF) [10] are features based on SIFT with several simplifications in calculation process to get features that have comparable distinctiveness but are less computationally demanding and have smaller descriptors. SURF are detected using Hessian matrix as SIFT, but the input of the Hessian matrix is calculated using the approximation of Gaussian functions by box filters, see Figure 5b. The scale invariance of SURF is achieved by filtering the image using different sizes of box filters for each SURF scale k.

SURF descriptor (shown in Figure 5a) is calculated from a square of $20k \times 20k$ and this square is further divided into 4×4 subregions. Then, each subregion is represented by the gradients as subregions of the SIFT descriptor. After that, the gradients are weighted by the box filters that are used as approximation of Gaussian function.

SURF are invariant to the image scale, rotation, and change of the viewpoint. The main advantage of SURF is that they are less computationally demanding than SIFT features and they have equal distinctiveness. Matching of SURF is performed using the Euclidean distance. Calculation of the Euclidean distance is computationally demanding, and the SURF descriptors are still relatively large, but it is possible to use the same tricks of decreasing the memory and computational requirements as can be used for SIFT.



(a) SURF descriptor

(b) Box filter approximation of Gaussian function

Figure 5: SURF descriptor construction (retrieved from [10])

2.2.3 BRIEF

Binary independent elementary features (BRIEF) [11] were designed to get features that are less computationally demanding and have smaller descriptors than SIFT features or SURF. BRIEF is described by its descriptor, which is a binary vector and can be used with any feature detector. Calculation of each descriptor's bit is based on the intensity comparison of two pixels in the neighborhood of the image feature coordinates. Two examples of pairwise comparisons are depicted in Figure 6.

BRIEF described by binary descriptors can be easily matched using the Hamming distance. When the Hamming distance between the two descriptors is small, there is a high probability that both descriptors belong to the same image feature. The great advantage of the Hamming distance in comparison to the Euclidean distance is that, the Hamming distance can be computed much faster[1]. Therefore, BRIEF is less computationally demanding than SIFT or SURF.

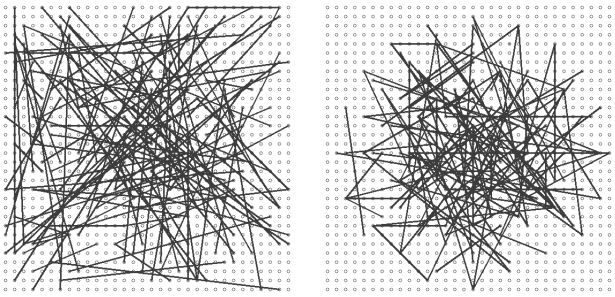


Figure 6: Examples of pairwise comparisons used in BRIEF descriptors (retrieved from [11])

2.2.4 ORB features

Oriented FAST and Rotated BRIEF (ORB) [12] is an image feature that uses improved FAST feature detector [13] and improved BRIEF descriptor [11], where the FAST stands for Features From Accelerated Segment Test. FAST detect features using comparisons of intensities of the pixels in the circle around the central pixel, see examples in Figure 7. If all n contiguous pixels in the circle have intensity higher or lower than central pixel, then the central pixel is considered as a new feature. There are two major improvements which makes ORB from FAST. FAST is not scale invariant; so, the first improvement provides the scale invariance by computing FAST from the same image at different image scales. The second improvement is based on the calculation of *intensity centroids* [14] to make FAST rotation invariant.

Calculation of the ORB descriptor is based on BRIEF comparisons. Moreover, different patterns of BRIEF comparisons are used for different orientations of the detected feature. This approach is used in order to make the BRIEF descriptor rotation invariant.

[1]Modern processors usually have a special instruction for calculation of the Hamming distance.

The main advantage of ORB features is that they can be computed faster than SIFT or SURF. Appeareance of the detection features as ORB, SURF, and SIFT detection is depicted in Figure 8.

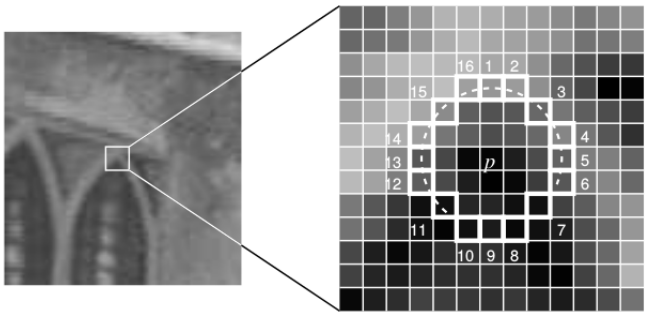
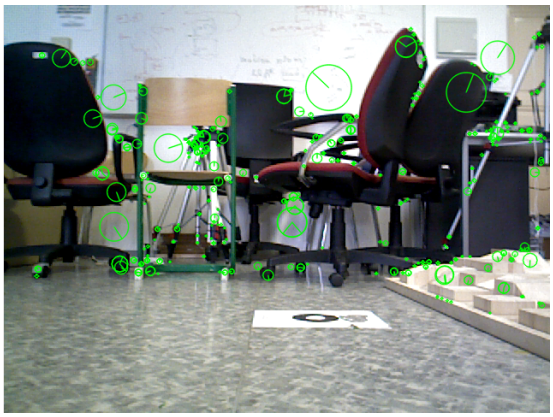
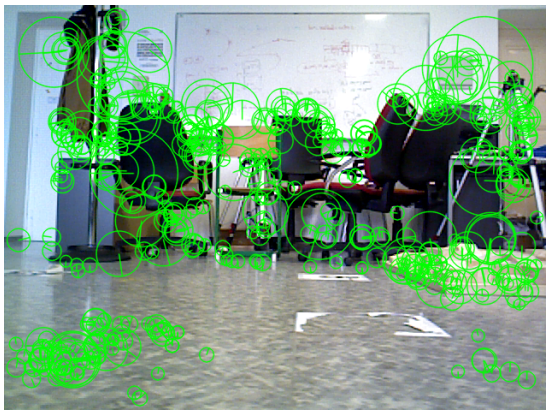


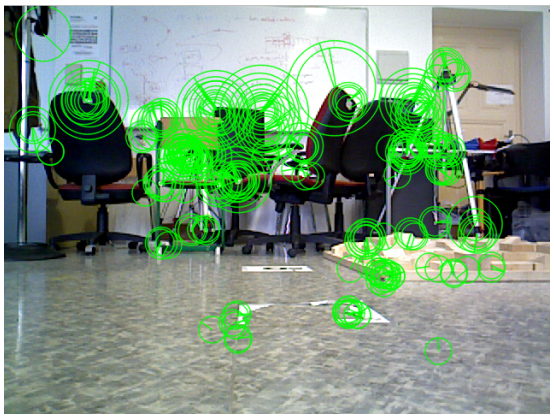
Figure 7: Example of FAST detector (retrieved from [13])



(a) SIFT



(b) SURF



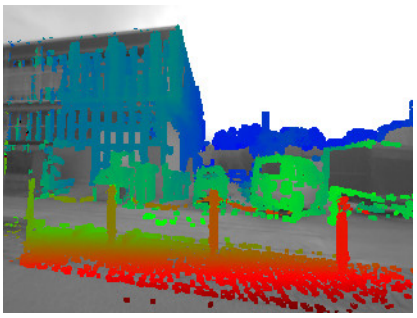
(c) ORB

Figure 8: Different features detected by RGB-D SLAM. Noteworthy is difference induced by the appearance nature of the ORB detector in contrast with the behavior of SURF and SIFT, which detect special regions.

Part 3
Visual simultaneous localization and mapping

Visual SLAM systems take an advandage of rich information about the captured environment contained in an image data to provide a reliable localization of a mobile robot. SLAM systems can be divided by the way the image data are processed to full/online, sparse/dense, and direct/indirect.

Full SLAMs estimate the whole robot's path and are capable to recalculate the path when a loop closure is detected. Whereas online SLAMs seek to estimate only the current camera pose. The advantage of full SLAMs is that they can improve quality of the localization by means of improving the underlying map of features whenever the robot revisits the same place. On the other hand, online SLAMs usually have lower hardware requirements as it is supposed they are running on-line and on-board of the robot computational resources.



(a) Pixels used for depth map costruction by LSD-SLAM (retrieved from [15])

(b) Image features detected by ORB-SLAM

Figure 9: Use of information by semi-dense method and sparse method

The amount of information taken from an image can be utilized to further clasify a SLAM systems as sparse or dense. SLAM systems clasified as sparse use just a small set of image pixels to construct a coarse map of the environment and calculate the camera's pose. Examples of sparse SLAMs are for example ORB-SLAM [2] or RGB-D SLAM [3]. On the other hand, dense SLAMs use all or most of pixels in an image to construct a detailed map of the environment and compute the camera's pose. The advantage of dense SLAMs is that their underlying map contains more details about the robot's surrounding and the map can be used also for other purposes, e.g., collision avoidance. However computational demands of dense methods are really high and these methods

usually require image processing on a graphic processing unit (GPU). In Figure 9b, it is shown an example of detected keypoints by the ORB-SLAM, and in Figure 9a an example of the selected pixels by the semi-dense method LSD-SLAM [15] is shown.

The last way of categorizing SLAM systems is based on the way of using the image data, which can be done directly or indirectly. Direct SLAM methods use intensity of pixels to compute a depth of the scene directly. The indirect SLAM methods compute a special representation of the selected pixels and their neighborhood (image features) and then a depth of each individual feature is calculated.

A more thorough description of the three feature-based visual SLAM systems that have been selected for the deployment and benchmarking on a hexapod walking robot as a part of this thesis follows. All the selected SLAMs are based on the visual odometry in discussed in Section 2.1 and provide localization of the robot in 6 Degrees of freedom (DOF). The studied SLAM systems use different types of features: SIFT [9], SURF [10], BRIEF [11], and ORB [12], already introduced in Section 2.2. Source codes of all the presented SLAM systems are open source and publicly available[2]. The evaluated SLAM systems are detailed in the next sections.

3.1 ORB-SLAM

ORB-SLAM [2], [16] is a feature based SLAM system, which enables to use three different sensors: RGB-D camera, monocular camera, and stereo camera. ORB-SLAM architecture (depicted in Figure 10) is divided into three parts that run in parallel: 1) tracking, 2) local mapping, and 3) loop closing.

Tracking provides the camera localization for each new frame and it is responsible for a keyframe selection. Camera localization is based on the visual odometry enhanced with an optimization using the bundle adjustement [17] and it works as follows. At first, features in a new frame are detected. Then, they are matched with features from the previous frame. Feature matching is accelareted by tracking of features, which means that each feature is matched with features in a close neighborhood of the feature's presumed position in the next frame. Feature tracking accelerates matching because each feature does not have to be matched with all features detected in the frame before. Afterwards, a new frame is compared with other frames, that supports reobservability of the previously seen features. These frames are found in the covisibility graph, see Figure 11. If the matching phase fails, then ORB-SLAM employs a module for the place recognition to relocalize the camera. Tracking is also in charge of a keyframe selection which happens when the frame contains sufficient quantity of new features. Keyframe is the frame which contains sufficient amount of new information to be stored. The main reason of Keyframe selection is to keep just the most important information about the environment.

[2]ORB-SLAM is available at `http://webdiis.unizar.es/~raulmur/orbslam/`, RGB-D SLAM is available at `http://felixendres.github.io/rgbdslam_v2/`, and S-PTAM is available at `https://github.com/lrse/sptam`

Local mapping processes a keyframe detected by tracking, using bundle adjustment to optimize the local map. Local mapping also extracts the most important features from the keyframes and put them into the global feature map, which is used when tracking is lost.

A loop closure detection is based on detecting sets of features that were already observed, which is used to improve the precision of the localization [1]. ORB-SLAM detects loop closures for each individual keyframe. When a loop closure is detected, the algorithm calculates error accumulated in the loop. Then, it corrects the graph of camera poses and also merges duplicated points and features in the map.

The whole system uses ORB features [12], which are classified as close or far based on the distance from the camera center. Distances of the close features can be reliably triangulated to be used for the visual odometry, especially for the calculation of the camera translation. On the other hand, a distance of far features cannot be reliably estimated, but these features are usually suitable for, e.g., the loop closing.

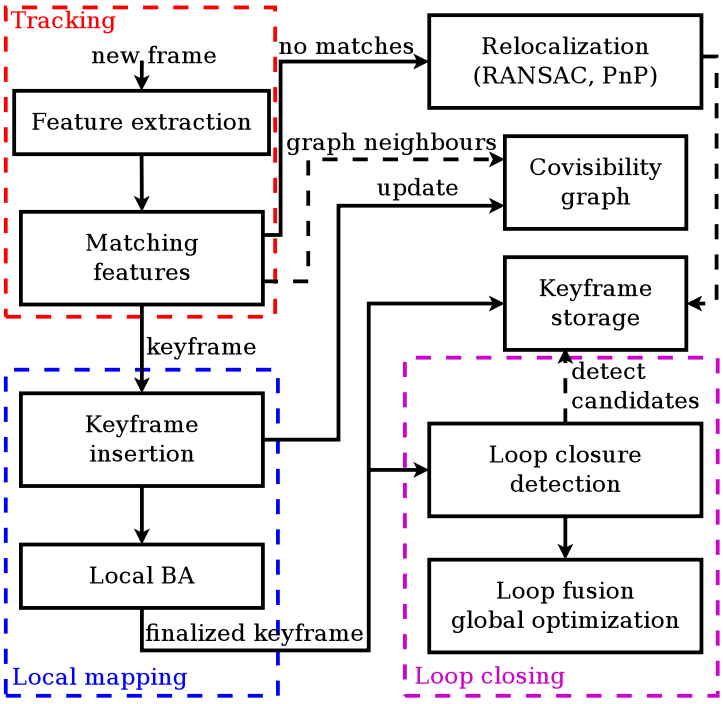


Figure 10: ORB-SLAM overview

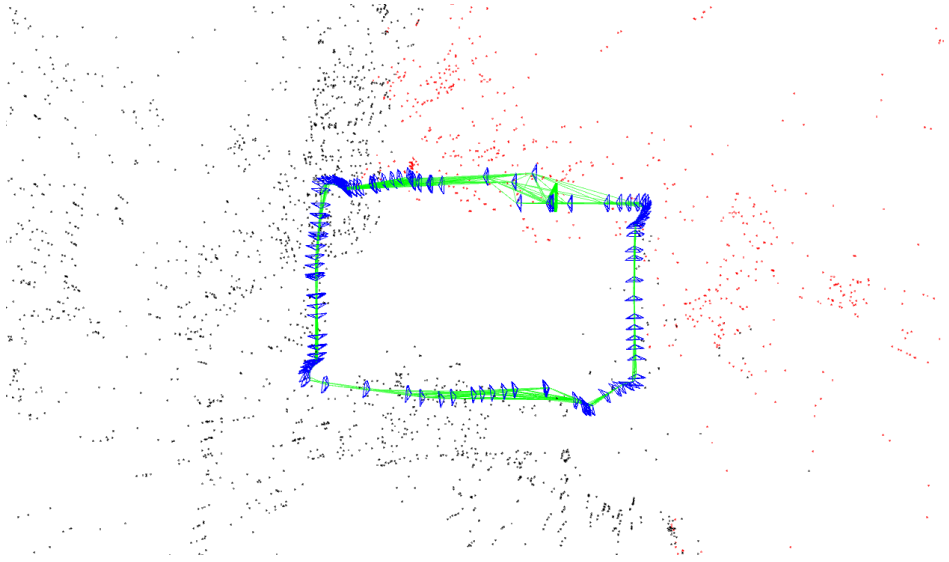


Figure 11: ORB-SLAM covisibility graph and feature map

3.2 RGB-D SLAM

RGB-D SLAM [3], [18] is the second version of the SLAM system designed to be used with an RGB-D camera like Microsoft Kinect or Asus Xtion that are both based on the Prime sensor chip. RGB-D SLAM is overviewed in Figure 12, where it can be seen that the architecture is divided into two main parts: the frontend and backend. The frontend is in charge of the feature detection, descriptor extraction, matching features, and calculating of geometrical transformations. The backend is responsible for the pose refinement and pose graph optimization.

When a new RGB image and corresponding depth image are captured, feature detection and feature description is performed first. RGB-D SLAM uses OpenCV [19] implementation of feature detectors and feature descriptors; so, it is possible to straightforwardly use different feature extractors, e.g., ORB, SURF, SIFT. It is also possible to use SIFT optimized for calculation on GPU, but in this thesis, it is supposed the SLAM system will be deployed on a mobile robot with an embedded computational platform of the type Odroid XU4, which does not feature an OpenCL enabled hardware. Authors of the RGB-D SLAM propose to use ORB features, because of their performance. However, in scenarios with a hexapod walking robot, it has been reported that features with higher distinctiveness provide better performance [20], [21]. Therefore, during all experiments performed in this thesis, SURF features (in both detectors and descriptors) were used.

After the feature extraction, frames are selected and compared with the current frame to find a large loop closure or just to find correspondences between following frames. There are three sets of frames selected for the comparison. The first set contains n_p frames that directly precede the current frame. The second set contains frames in the graph neighborhood of the first set (includes the previously detected loop

closures). Moreover, the last set includes n_r frames that are chosen randomly to detect new large loop closures.

After the selection, features detected in the current frame are matched with the features of all the frames from the selection. Then, a camera pose is estimated by the RANSAC algorithm [22]. The pose estimated by the RANSAC algorithm is then refined using a bundle adjustment and it is added to the pose graph. When a loop closure is detected, the whole pose graph is optimized.

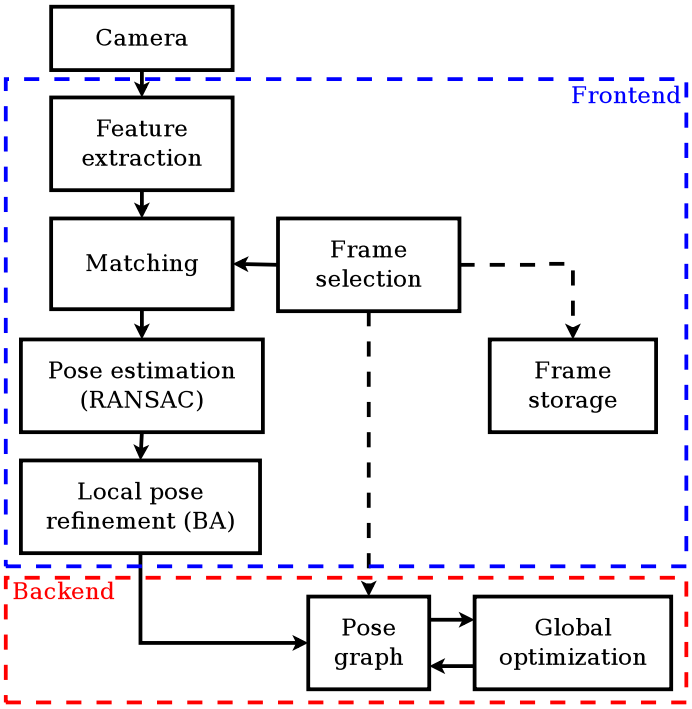


Figure 12: RGB-D SLAM overview

3.3 S-PTAM

Stereo Parallel Tracking and Mapping (S-PTAM) [4], [23] is a SLAM system mainly designed to be used with a stereo vision sensor system. Its architecture is divided into three parts that run in parallel: extraction, tracking, and mapping. Figure 13 shows the S-PTAM operation.

The extraction is in charge of the keypoint detection and the descriptor computation. Keypoints are detected using the Good features to track (GFTT) detector [24] and described by the BRIEF [11] feature descriptors. The used combination of feature detector and feature descriptor is implemented using OpenCV [19]; so, it is possible to easily use different image features implemented under OpenCV. However, the selected combination of GFTT detection and BRIEF description provides the best performance, according to [8]. Notice, image features are detected in one image, and then S-PTAM

atempts to detect the same features at corresponding epipolar(s) of the second camera image.

Tracking works similar as in ORB-SLAM. It is in charge of feature matching, and estimation of the camera pose. When a new pair of images is captured, all image features are then projected onto the plane with features that were detected in the former frames. The projection is based on a prediction of the current camera pose. Feature matching is performed using the Hamming distances between not-matched feature and features in the close neighborhood in the plane. Therefore, the computational demands depend on the quality of the camera pose prediction. S-PTAM uses a velocity motion model, or a model based on the Extended Kalman filter (EKF) [1] enhanced with IMU measurements to predict the camera pose. In this thesis, S-PTAM is used with the stereo camera TARA, which provides IMU measurements; so, it would be possible to use a model based on the EKF, which is more suitable for walking robots. Since the motion of the wheeled robots is generaly smoother than the motion of walking robot, the estimation of camera pose of a wheeled robot is much easier. However, only vision based SLAM systems without any aditional sensors are compared in this thesis; thus, S-PTAM that relies only on the velocity motion model was utilized.

The tracking phase also selects keyframes, which are frames that contain less than 90 % of points tracked in the previous keyframe. New keyframes that were detected during the tracking are used for the map construction.

Mapping inserts new points in the map and refines the camera position and the local map utilizing BA. Optimization is used only for a local map containing just nearby keyframes. The absence of the global map optimization prevents false loop closing.

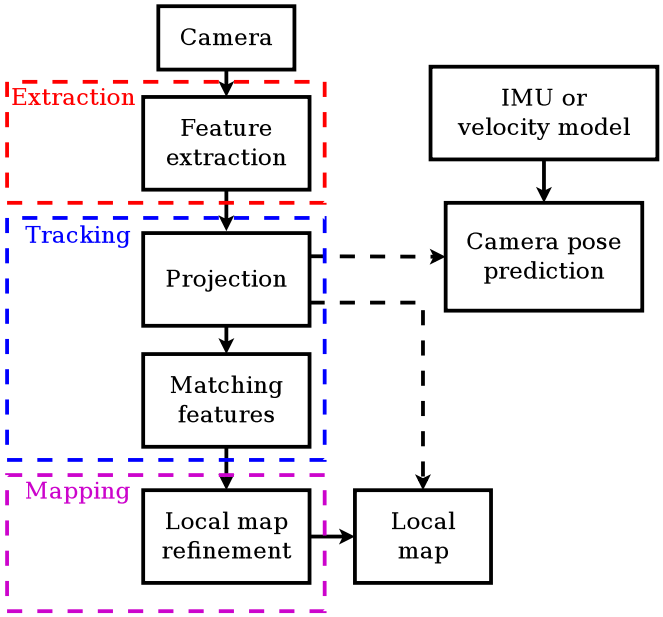


Figure 13: S-PTAM overview

3.4 Active localization

SLAM systems can use techniques like loop closures to improve the localization, but when the robot does not perform actions that enable usage of these techniques, the localization cannot be improved. This is the principle of passive localization when the localization system just processes data from sensors, but it cannot affect the robot's actions. On the other hand, there is active localization approach [25], which main principle is to affect the robot's actions to improve the robot's localization.

Experiments performed in this thesis and reported in Section 5 were designed in order to compare the performance of several SLAM systems and also to find out, which robot's actions significantly affect the accuracy and reliability of the robot's localization. Some of these actions can be used to improve localization of the robot using the active localization approach in a similar way as in [26], [27].

Part 4
Evaluation methodology

All SLAM systems described in Chapter 3 (i.e., ORB SLAM, RGB-D SLAM, and S-PTAM) provide localization of the robot and a map of the robot's surroundings. One of the ways to compare SLAM systems is to compare just the accuracy of the localization provided by the SLAM systems in the same scenario. It is also possible to evaluate the quality of the constructed map. However, as it is noted in [18], the quality of the map depends on the accuracy of the estimated trajectory, and thus it may not be necessary to evaluate SLAM systems using the created maps.

The accuracy of SLAM can be measured by comparing the whole estimated trajectory of the robot with the reference trajectory (the ground truth). The ground truth should be enough precise for the comparison, therefore an external localization system April [6] with the centimeter precision was used to capture the ground truth during all experiments reported in this thesis. Nevertheless, the evaluation of the estimation of the trajectory accuracy requires metrics that are capable to compare the quality of two estimations as well as to correctly express an error of the estimated trajectory. Metrics suitable for the evaluation of trajectory estimation are the Absolute trajectory error (ATE) and relative pose error (RPE) [5]. These metrics were also used in similar approaches of SLAM systems evaluation in [28] and [20] as they are presented in this thesis.

This chapter describes the evaluation process used for the evaluation of experimental data. Section 4.1 depicts the process of establishing a reliable ground truth. The utilized evaluation metrics are detailed in Section 4.2.

4.1 Ground truth

For the ground truth construction, a vision based absolute localization approach has been used. The used absolute localization works as follows. At first, a video with the robot's motion was captured by one Logitech c920 HD webcam with a resolution of 1920 × 1080 px at approximately 30 FPS that has been attached above the robot operational work space. For easier recognition of the robot, the AprilTag [6] has been attached to the robot. After that, the video has been captured and processed frame-by-frame using the April detector [6] to get the trajectory of a particular marker (AprilTag), which has been attached to the robot.

The great advantage of the motion capture system based on the April detector [6] is that only one marker (AprilTag) is needed to get estimation of the full 6 DOF position of the robot. It is also possible to use different vision-based motion capture system like Whycon [29], which uses different, rotation invariant markers. However, it is necessary

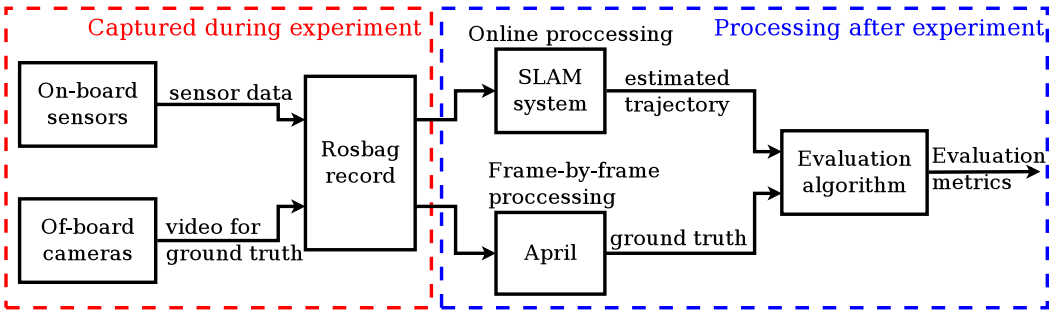


Figure 14: Schema of the evaluation process

to attach more than one marker to get full 6 DOF position, because of the rotation invariance of the utilized marker in the Whycon [29] system, which has been found impractical for the purpose of this thesis.

A comparison of the estimated trajectory with the ground truth requires timestamps for all frames of the ground truth. But standard mjpeg video does not provide reliable timestamps for each frame of the video. Therefore, all videos used for ground truth construction have been captured using ROS framework [30] with rosbag. The ROS's rosbag is capable of capturing all data coming from the sensors as individual messages and save them to the bag file. The advantage of this framework is that all messages are timestamped. The timestamp corresponds to the moment when the message was published. During all the experiments described in Section 5, data from sensors attached to the robot and video stream for the ground truth construction have been saved in one bag file, which ensures no time shift between the ground truth and estimated trajectory.

4.2 Evaluation metrics

The output trajectory of the SLAM system and corresponding ground truth are represented by timestamped 6 DOF poses. As all of the sensory data are recorded into a single bag file on a single computer, the timestamps are globally consistent. Nevertheless, the individual sensors provide the data with a different frequency, which has to be synchronized before the evaluation. The synchronization consists of two phases. The first one extracts the corresponding poses of trajectory estimate and the ground truth based on the timestamps. The second one seeks a rigid transformation between the trajectory estimate and the ground truth to synchronize the coordinate systems which are, in general, different.

The first phase of the synchronization can be done by finding pose with the nearest timestamp of the ground truth for each pose of the estimated trajectory as it is proposed in [5]. However, in this thesis, we propose to use linear interpolation of the ground truth poses to improve the reliability of the evaluation metrics and better synchronization of the trajectories with respect to time domain. The linear interpolation of the ground truth consists of the linear interpolation of the position and linear quaternion interpolation (LERP) as an interpolation of the orientation. Note, whereas the interpolation of

the position is straight-forward, the interpolation of the rotation is much harder to assess. Therefore, we use the linear quaternion interpolation and normalize the resulting quaternion to represent a valid rotation.

The second phase of the synchronization consists of estimation of translation and rotation between the time synchronized trajectory and ground truth. The rigid transform was found by collocating the starting points of the trajectories and then seeking the rotation that minimizes the root mean squared error between the estimated trajectory and the ground truth.

Each pose of the estimated trajectory and corresponding pose of the ground truth have to be represented by the SE(3) matrix. The SE(3) matrix is representation of the robot's position by the matrix of size 4×4, which can be divided into the orientation represented by a rotation matrix R (of size 3×3) and the column matrix of the translation T

$$P_{SE(3)} = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix}. \tag{1}$$

After all the mentioned preparations, it is possible to compute ATE and RPE as follows. ATE [5] is calculated using the equation:

$$F_i = Q_i^{-1} P_i, \tag{2}$$

where Q_i represents an ith point of the ground truth by the SE(3) matrix[3], and P_i represents ith point of the estimated trajectory. ATE represented by the SE(3) matrices F_i can be divided into the translational part of ATE F_{itrans} and the rotational part of ATE F_{irot}, as it was used in [20]. The translational part of ATE measures distances between the estimated poses and poses of the ground truth. Similarly, the rotational part of ATE measures an error of the robot's orientation.

RPE [5] can be calculated using the equation

$$E_i = (Q_i^{-1} Q_{i+\Delta})^{-1} (P_i^{-1} P_{i+\Delta}), \tag{3}$$

where Δ defines the size of the fixed frame interval. RPE represents a local drift of the estimated trajectory. When $\Delta = 1$, then E_i expresses a drift of the visual odometry. RPE can also be used to evaluate conformity between shape of the estimated trajectory and shape of the ground truth, which can be done using $\Delta > 1$. In this thesis, the conformity of the evaluated trajectory was measured by chosing Δ to achieve distance between frames approximately 0.5 m, which is the size of the used robot. In the following text, the translation part of the RPE where Δ corresponds with the distance of 0.5 m is referenced as RPE_{tl}.

RPE represented by the SE(3) matrices E_i can be divided using the same principle as ATE into E_{itrans} and E_{irot}. Results of the trajectory evaluation detailed in Section 5 are expressed utilizing average values ATE_t, ATE_ϕ, RPE_t and RPE_ϕ computed from $|F_{itrans}|$, $|F_{irot}|$, $|E_{itrans}|$, and $|E_{irot}|$.

[3]Representation of poses by the SE(3) matrix can be expressed as a transformation of the standard basis of the space $\mathbb{R}^3$ from the begining of the coordinates.

Part 5
Experimental results

In this chapter, experiments on the localization of a real hexapod crawling robot are described together with the description of the different experimental setups and the robot itself. During all experiments, the same robot platform was used in two different configurations. These configurations differ in the used main sensor for the localization. The first configuration is the RGB-D camera and the second configuration is with the stereo camera Tara.

5.1 Robot's description

Experiments were performed on the off-the-shelf hexapod robot based on Phantom X Mark II. The used hexapod robot platform consists of six legs and the robot's body with the robot's control unit and a camera. Each leg of the robot is composed of three Dynamixel servos AX-12A with the inbuild sensor of angular rotation, which can be used to detect a contact point of the robot's leg with the ground, see Section 5.1.1. Due to the 3 actuators per each leg, it is possible to set a position of each leg's endpoint at 3 degrees of freedom (DOF). The robot servos enable to carry about 2.5 kg of the robot's mass and about 1 kg of additional load. The robot loaded with all the necessary equipment for experiments presented in this thesis weights about 2.7 kg. Battery, which powers the robot is a Li-pol battery pack (3s, 4 Ah), which provides enough energy to power the robot for about 90 minutes. The maximal distance, the robot can traverse by a single battery pack depends mostly on the type of the motion gait [31]. Servo motors of the robot are controlled by the robot's control unit Odroid U3+, which is an ARM Cortex A9 based quad-core embeddded computer. A PS2-like controller communicating over wireless protocol is used for the remote control of the robot.

5.1.1 Motion gaits

Locomotion of the walking robots can be based on two types of control; the first one is using a pattern of movement (motion gait). The second type uses planning of each foothold. Motion gaits, with a pattern of movement can be usually used without the need of any additional sensors. On the other hand, planning of foothold placement is more suitable for tough terrains like stairs. However, it is computationally demanding and the whole motion of the robot is very slow, due to the alternation between the sensing and planning. Therefore, the pattern based motion gait is utilized for the robot locomotion in this thesis. However, the motion gait if further featured by the ability to traverse uneven terrain.

The utilized approach to traverse difficult terrain with the hexapod robot is the adaptive motion gait introduced in [32], which is based on a motion pattern enhanced with a feedback from the actuators that enables to detect when a leg touches the ground. The adaptive motion gait allows the robot to move faster with lower computational cost than with the foothold placement planning approach.

Motion gaits of a hexapod walking robot can also be divided into groups according to the number of legs that are in the support phase. If the terrain is flat, it is possible to use motion gait without any feedback (default gait), which enables the robot to move faster. The faster locomotion of the hexapod robot yields, the robot consumes less energy per traversed distance [31]. When a simple gait without a feedback is incapable of traversing uneven terrain, it is possible to use adaptive gait, which is slower, and therefore, less energy efficient, but it enables the robot to traverse much more difficult terrain. If a terrain is extremely difficult, e.g., there are just a few places, where the robot can place its legs, then planning of each foothold placement [33] can be more suitable choice. In this thesis, the adaptive motion gait together with the default gait were utilized for the data collection in the experimental evaluation of the SLAM systems. The adaptive motion gait was used in all experiments in Section 5, whereas the default gait was used only for experiment presented in Section 5.2.4.

5.1.2 Sensors

SLAM methods used in this work for the localization of a hexapod robot use different types of sensors: RGB-D SLAM V2 uses an RGB-D camera, S-PTAM uses a stereo camera and ORB-SLAM2 enables to use an RGB-D camera, stereo camera, and also a monocular camera.

5.1.2.1 RGB-D camera

The RGB-D camera provides standard an RGB color image and also depth image which contains distances from camera center to points that correspond to each pixel of the color image. For the experiments with SLAM algorithms, the RGB-D camera Asus Xtion Pro (newer version) was used, which is more suitable for mobile robotics than, e.g., Microsoft Kinect which is heavier and requires an extra power supply. Resolution of the color image provided by the camera is 640×480 px with the framerate of 30 FPS. Distance measurements (depth image) are provided by an infrared sensor, which works as follows. A source of the infrared light emits a special pattern of the infrared light. Then, the pattern is reflected from objects in front of the camera. Different distances between the objects and camera defect the infrared pattern. Next, the infrared sensor captures a reflected pattern and computes the distances from the comparison of the reflected pattern and the reference pattern. Note, the Asus Xtion uses a camera sensor which has not a global shutter and the RGB and depth images are not synchronized. The absence of the global shutter induces a geometrical distortion of an image, especially when the robot turns fast. Unsynchronized RGB and depth image results in wrong

estimation of the image depth when the robot moves fast. Henceforth, the estimation of feature's depth in the environment is less precise which results in an overall decrease of trajectory estimate precision provided by SLAM system.

5.1.2.2 Stereo camera

The stereo camera consists of a camera pair with two parallel optical axes at the defined distance called the baseline. The main advantage of the stereo camera is the ability to compute a distance between the camera and observed objects directly from a pair of images using triangulation. Therefore, some SLAM systems, e.g., [4], use stereo cameras rather than monocular cameras. The ability to directly estimate a position of each point of the map from just one measurement is a great advantage for initialization of the map. The initialization of the map using stereo camera is simple. It consists only of the insertion of the first pointcloud.

The stereo camera used in all practical experiments in this thesis was the E-con systems Tara [34]. Tara was designed for computer vision applications and for mobile robotics, which puts specific demands on the used sensors. One of the most important features of the sensors is the ability to capture all the image pixels at the same time. For this reason, Tara features the image sensors MT9V024 with the global shutter. The global shutter enables the camera to capture consistent images without artifacts even if the camera or object in the field of view moves relatively fast. Another important feature of Tara is synchronized capturing of images by both camera sensors, which improves the reliability of the triangulation. The next property, which makes Tara a good choice for the mobile robotics, is an in-built IMU. Tara's IMU is capable of measuring speed and acceleration of the camera in three Degrees of freedom (DOF). IMU, integrated directly in stereo camera, is beneficial especially for small robots. A disadvantage of Tara is that both cameras are sensitive to infrared light; so, it is impossible to directly use Tara together with the Asus Xtion because of the infrared pattern affects Tara's sensors. Properties of Tara as well as properties of Asus Xtion are depicted in Table 1.

5.1.2.3 Monocular camera

Although, it is beneficial to use a stereo cameras for SLAM systems, monocular cameras are smaller and more common not only in mobile robotics. Thus, they enable SLAM system to be deployed in different scenarios, e.g., for localization in buildings utilizing smartphones [35].

On the other hand, initialization of map using a monocular camera is much harder than initialization using stereo camera because the scale of observed scene is unknown. There are several ways to solve the problem with initialization of the map, e.g., bootstrapping problem [36]. The easiest way is to use a prior knowledge of the initial scene, e.g., put an object of known proportions in the scene. However, after the initialization of the map, the scale drifts, and therefore, a less precise results can be expected. For the evaluation purposes of a monocular SLAM system, one of Tara's cameras has been used.

Table 1: Properties of the used sensors

Sensor	Asus Xtion Pro	Stereo camera Tara
Resolution [px]	640 × 480	752 × 480
Framerate [Hz]	30	60
Global shutter	no	yes

5.2 Experiments

Localization provided by different SLAM systems has been evaluated for four different setups. In each setup, multiple trials (dataset) were captured by the hexapod robot in one or both possible configurations: with the RGB-D camera and with the Stereo camera (shown in Figure 15). Each captured trial was processed online[4] by the SLAM algorithms described in Section 3; so, there are multiple trajectory estimates for each setup. The only trials processed by ORB-SLAM at the framerate of 60 FPS were not processed online. Afterwards, all trajectory estimates have been evaluated using metrics described in Section 4.2. In all the experimental setups, the hexapod robot has been guided by a human operator.

It is also possible to evaluate the accuracy of SLAM methods using publicly available datasets like, e.g., Kitty dataset [37] or TUM dataset [5]. However, these datasets usually do not provide trials captured by both sensors (the RGB-D camera and stereo camera) and are also captured from a wheeled vehicle, whose motion significantly differ from the hexapod walking robot studied in this thesis. The particular experimental setups are further described in the following subsections.

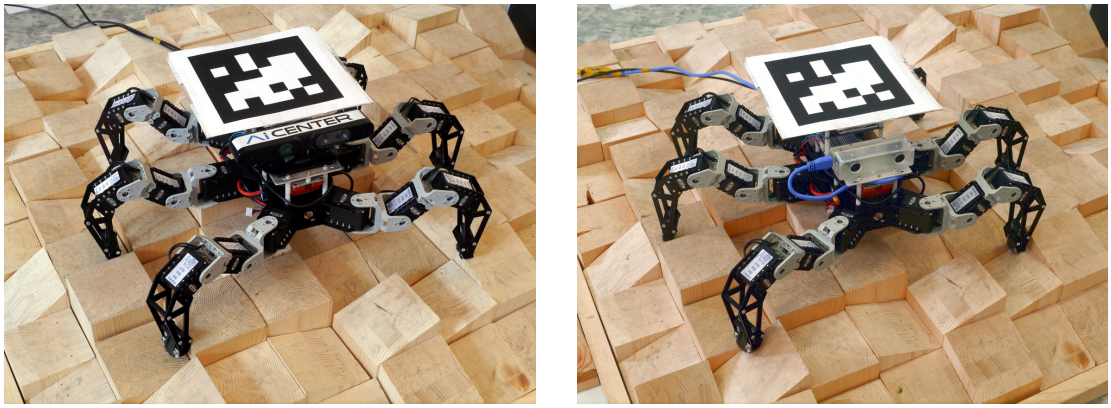


Figure 15: Hexapod robot equipped with Asus Xtion and with Tara.

[4]For the processing was used machine with 3.4 GHz Intel Core Xeon octa-core processor, 16GB RAM. However, not more than 2GB of RAM were used in the most demanding cases.

5.2.1 Setup 1: Planar Surface

When the hexapod robot walks on a planar surface, it may look like that the motion of the robot's mass center is similar to the motion of a wheeled robot, but there is a great difference in the smoothness of the movement. Whereas a wheeled robot moves smoothly, on a planar surface, crawling of the hexapod robot makes its movements non-smooth.

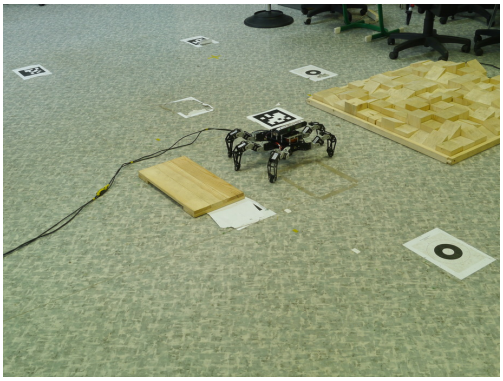


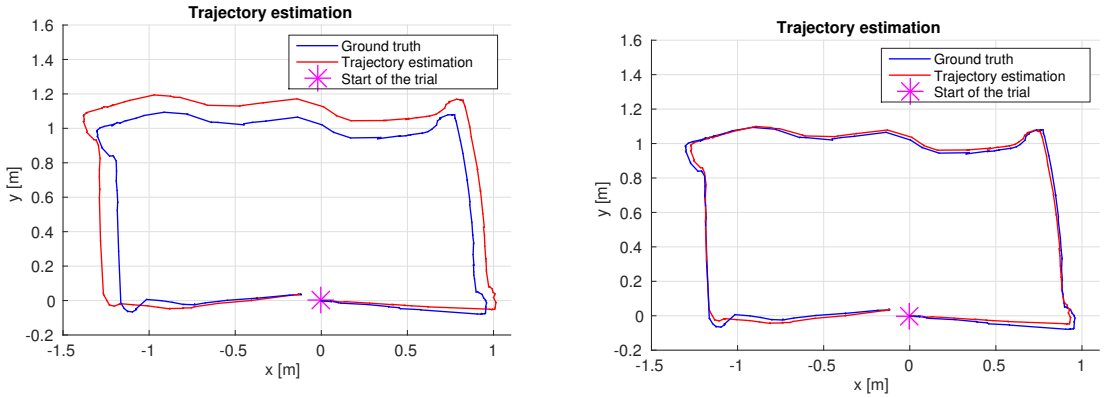
Figure 16: Experimental setup with planar surface.

In the Setup 1, the robot walks only on a planar surface in a space surrounded by many different objects at various distances, which provide many features for visual localization (see Figure 16). The dataset contains ten trials where the robot is equipped with the RGB-D camera and ten trials for the stereo camera. In all 20 trials, the robot walks approximately along the rectangular trajectory (about 7 m long). Therefore, it is possible to test SLAM algorithms in both close loop and open loop scenarios, as it was done in [28].

Evaluation process shows that trajectory estimates provided by both stereo methods (S-PTAM and stereo version of ORB-SLAM) were estimated at a wrong scale (see Figure 17). Therefore, all the trajectory estimates provided by stereo methods in experimental Setup 1 and 2 were used for the calculation of scale offset. Each trajectory was synchronized with corresponding ground truth (using minimization of RMSE between corresponding poses) while compensating the scale offset. The offset calculated for stereo methods is $k_{STEREO} = 0.921 \pm 0.024$. The offset k_{STEREO} was used for all trajectory estimations provided by both stereo methods. The same way of offset calculation was also used for RGB-D methods: $k_{RGB-D} = 1.009 \pm 0.054$; so, the compensation of the scale offset was not necessary.

Table 2: Trajectory estimation results for Setup 1

SLAM system	Close loop scenario							Open loop scenario						
	ATE_t [cm]	ATE_ϕ [deg]	ATE_e [cm]	RPE_t [cm]	RPE_{tl} [cm]	RPE_ϕ [deg]	No. of fails	ATE_t [cm]	ATE_ϕ [deg]	ATE_e [cm]	RPE_t [cm]	RPE_{tl} [cm]	RPE_ϕ [deg]	No. of fails
RGB-D SLAM	24.11	19.37	14.45	0.67	7.45	0.54	0	50.72	25.65	93.16	0.64	6.79	0.52	0
ORB-SLAM (RGB-D)	5.54	9.89	2.18	0.47	3.92	0.49	0	5.99	10.22	9.77	0.46	3.59	0.49	0
ORB-SLAM (Stereo)	4.41	5.78	4.26	0.38	2.74	0.53	1	4.86	6.40	9.04	0.36	2.60	0.54	1
S-PTAM (Stereo)	-	-	-	-	-	-	-	6.91	6.27	9.08	0.88	5.55	1.45	0
ORB-SLAM (Mono)	-	-	-	-	-	-	10	-	-	-	-	-	-	10



(a) Trajectory estimation at wrong scale

(b) Trajectory estimation with scale offset compensation

Figure 17: Scale offset compensation for trajectory estimations provided by stereo methods

All 10 trajectories estimated by each SLAM system were evaluated using the metrics described in Section 4 and then medians were calculated for each trajectory error. Incomplete trajectory estimates and trajectory estimates with false loop closures are counted as failed and excluded from the evaluation. The results are summarized in Table 2 and Figure 18.

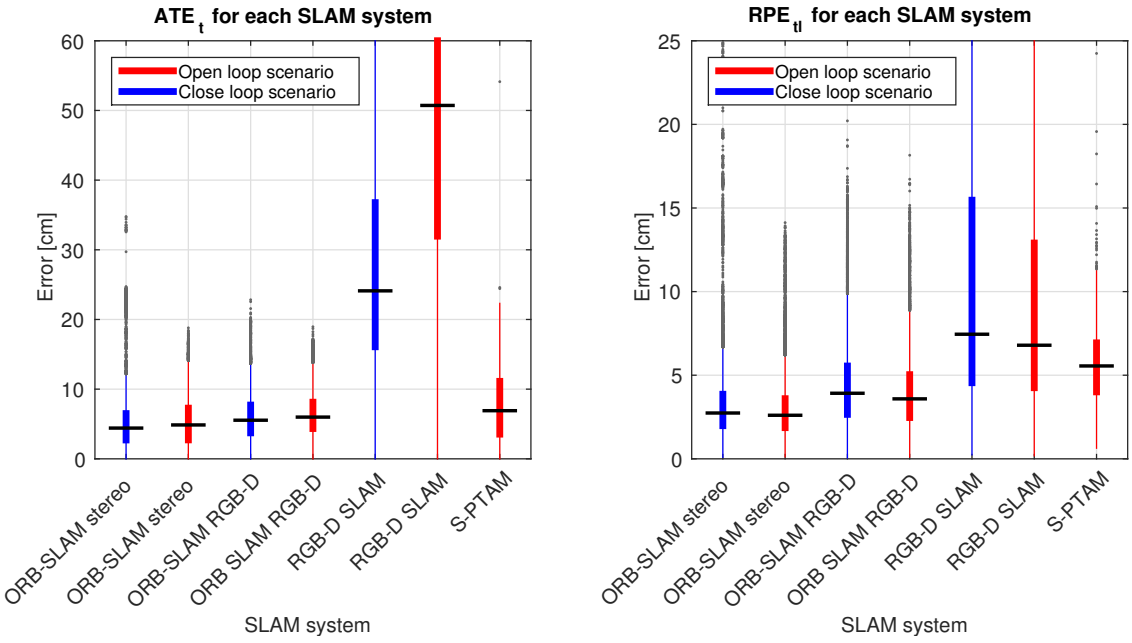


Figure 18: Absolute and relative error of SLAM systems in flat terrain scenario

5.2.2 Setup 2: Artificial Uneven Terrain

The ability of the hexapod robot to traverse a rough terrain induced experimental setup with a challenging uneven terrain. The experimental setup contains a corridor filled with wooden blocks with flat and slant tops, see Figure 19 and Figure 15. Slant tops of the blocks greatly affect the robot's motion, because the robot often slips down, when it tries to climb on the wooden blocks. An unpredictable slippage makes the robot's motion very rough. Slipping of the robot makes robot's motion almost unpredictable.

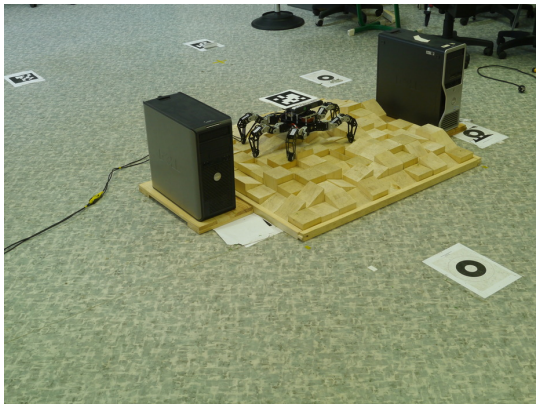


Figure 19: Experimental setup with artificial uneven terrain

The experimental setup is similar as Setup 1 described in Section 5.2.1 except that the robot walked about 6 m on a planar surface and about 1.1 m in the corridor filled with the wooden blocks. The dataset captured during this experiment also contains ten trials with Asus Xtion and ten trials with Tara.

Table 3: Trajectory estimation results for Setup 2

SLAM system	Close loop scenario							Open loop scenario						
	ATE_t [cm]	ATE_ϕ [deg]	ATE_e [cm]	RPE_t [cm]	RPE_{tl} [cm]	RPE_ϕ [deg]	No. of fails	ATE_t [cm]	ATE_ϕ [deg]	ATE_e [cm]	RPE_t [cm]	RPE_{tl} [cm]	RPE_ϕ [deg]	No. of fails
RGB-D SLAM	21.96	19.07	19.13	0.79	8.98	0.54	1	42.81	24.20	101.38	0.72	8.16	0.50	0
ORB-SLAM (RGB-D)	7.93	12.84	5.39	0.48	4.98	0.47	0	7.59	13.05	7.43	0.48	4.27	0.46	2
ORB-SLAM (Stereo)	7.89	7.10	11.19	0.41	3.21	0.56	1	5.88	7.88	14.70	0.41	2.86	0.58	2
S-PTAM (Stereo)	-	-	-	-	-	-	-	7.50	6.55	9.80	0.95	5.39	1.49	0
ORB-SLAM (Mono)	-	-	-	-	-	-	10	-	-	-	-	-	-	10

Evaluation of trajectory estimation were performed in the same way as it was done in the previous experiment. The results shown in Table 3 and in Figure 20 indicate that method with generally the lowest trajectory error is stereo version of ORB-SLAM.

In one case, RGB-D SLAM failed to estimate the trajectory because it detects a false loop closure (depicted in Figure 21). The false loop closure significantly affect the trajectory error. ORB SLAM (RGB-D and stereo version) failed mostly at the corners of the trajectory where the ORB SLAM starts to track new features with insufficient speed which results in tracking failure. Monocular version of the ORB-SLAM failed almost every time at the first corner of the trajectory.

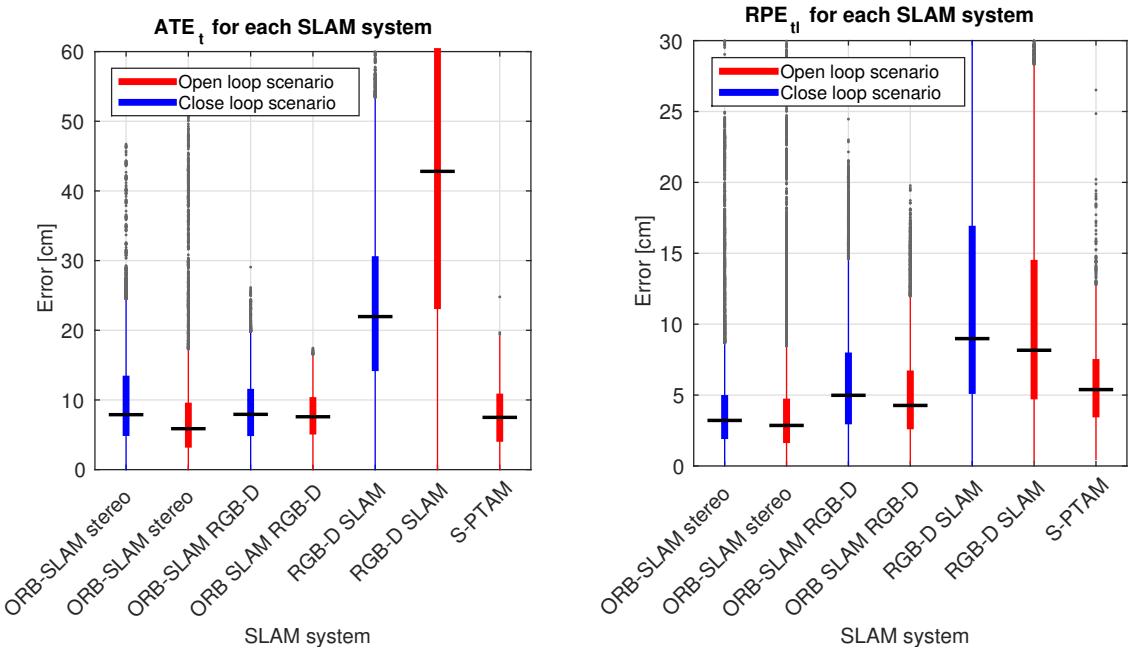


Figure 20: Absolute and relative error of SLAM systems in uneven terrain scenario

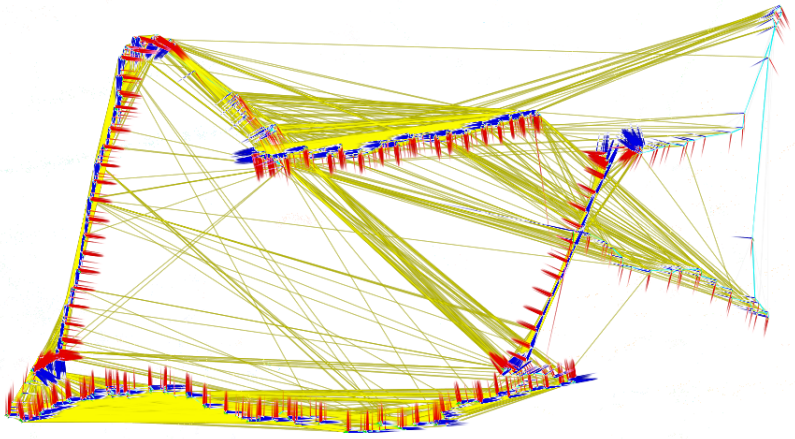


Figure 21: Pose graph after optimization applied on false loop closure by RGB-D SLAM. Yellow lines represent the established frame-to frame correspondences. Whereas the ones along the trajectory are correct, those between different parts of trajectory are incorrect and induce the false loop closures.

5.2.3 Setup 3: Test of Mono SLAM

In the previous setups described in Sections 5.2.1 and 5.2.2, the monocular version of the ORB-SLAM was unable to localize the robot in all trials. In most of the experiments, the localization failed in the first corner of the trajectory. Otherwise, there was at least a significant drift of the feature map's scale.

The main hypothesis, why SLAM with a monocular camera fails, is that ORB-SLAM

starts to track new image features only if it is able to estimate their distance from the camera center, which requires to see these features from multiple distances. However, when the robot turns with almost zero turn radius, new features are not added to the map as there is no lateral movement that would enable the triangulation. A special experimental setup has been prepared to test this hypothesis. An oval shape of the robot's trajectory, where the robot turns with a non-zero turn radius has been designed. This experimental setup also contains more objects at different distances from the camera at the start of the trajectory to make the map initialization easier. The dataset contains two trials captured by the robot equipped with the Tara camera. Each trial was processed using the stereo version of the ORB-SLAM and its monocular version utilizing only the left camera image. Both trials of the dataset were processed two times at 60 FPS in a close loop scenario. The results are summarized in Table 4.

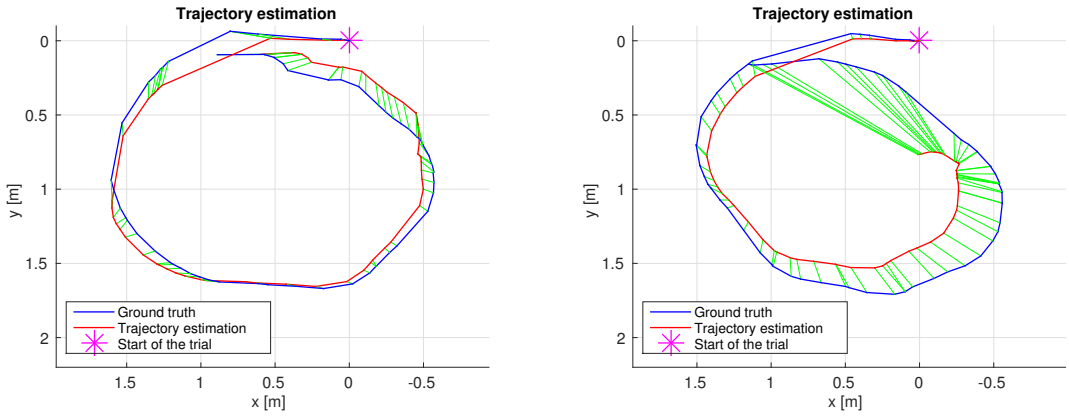
Table 4: Trajectory estimation results for Setup 3

	ORB-SLAM (Mono)				ORB-SLAM (Stereo)			
	Trial 1		Trial 2		Trial 1		Trial 2	
Run	1	2	1	2	1	2	1	2
ATE_t [cm]	10.37	-	7.57	18.79	2.54	2.88	2.63	3.73
ATE_ϕ [deg]	7.31	-	7.18	8.39	4.20	4.52	4.34	4.86
ATE_e [cm]	29.56	-	7.85	129.29	1.10	1.34	0.30	1.31
RPE_t [cm]	2.47	-	2.10	2.47	1.89	1.67	1.55	1.62
RPE_{tl} [cm]	10.93	-	6.76	14.11	4.20	4.84	4.22	4.28
RPE_ϕ [deg]	1.87	-	2.01	1.71	1.74	1.62	1.40	1.64
Note	LC	TL	TL	LNC	LC	LC	LC	LC

Special designation is used in Table 4 to express some additional information about trials; LC means the loop was closed, LNC means the loop was not closed, and TL means the tracking was lost. In Setup 3, the monocular version of the ORB-SLAM was able to estimate three of four trajectories. Results in Table 4 show that localization provided by stereo version of ORB-SLAM is in all cases more accurate than the localization provided by monocular version of the ORB-SLAM.

Results in Table 4 indicate that the monocular version lost tracking of features in two cases; during the second run of the first trial and during the first run of the second trial[5]. When the cases where ORB-SLAM does not lost tracking are compared, it can be concluded that Loop closing significantly improves the accuracy of the localization. This effect is induced by the drifted scale of the ORB-SLAM's map, which is shown in Figures 22b and 22a.

[5]In this case, ORB-SLAM covers most of the trajectory estimate using relocalization.



(a) First run of trial 1 (loop closed) (b) Second run of trial 2 (loop not closed)

Figure 22: Trajectory estimations by Monocular version of ORB-SLAM with marked ATE_t (green color)

5.2.4 Setup 4: Different Viewpoint Orientations

The last experimental scenarios have been designed to test the impact of the hexapod robot's heading on the accuracy of the localization. Note, the hexapod robot is an omnidirectional vehicle; so, it can move in any direction without rotation around the vertical axis unlike as, e.g., a car. The main idea of experimenting with the robot's orientation is to determine how walking directions affect the accuracy of the localization.

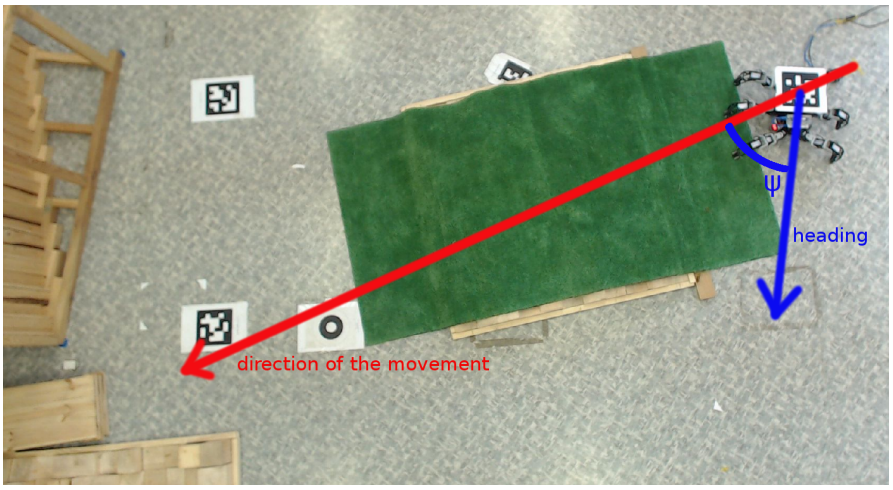
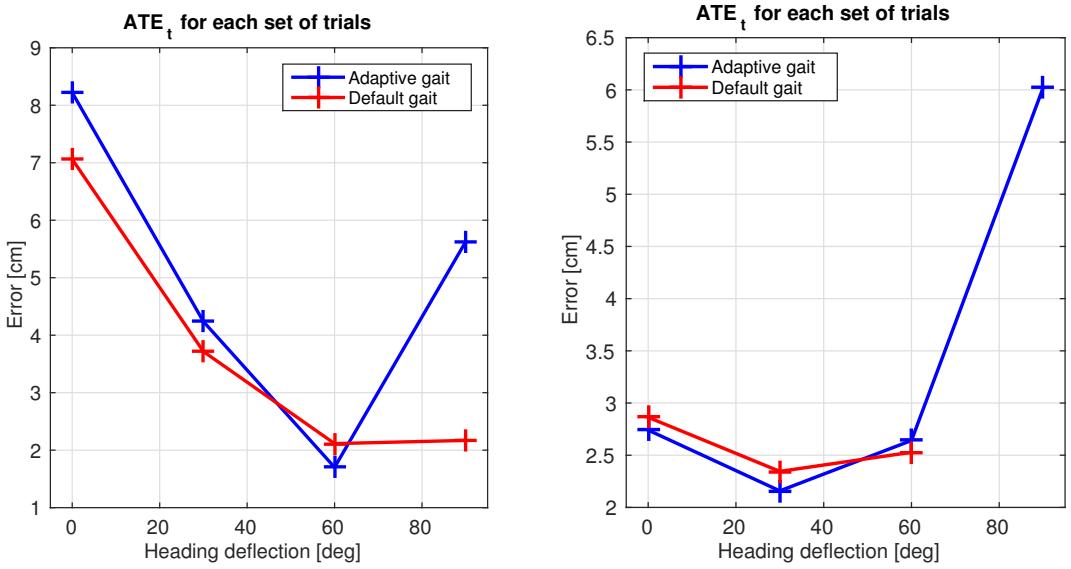


Figure 23: Experimental setup with different viewpoint orientations. The robot was guided from right to left along the marked red line with heading deflection ψ of 0,30,60 and 90 degrees respectively.

Therefore, the experimental setup contains a set of experiments with a non-zero angle ψ between the robot's move direction and the heading of the robot (the camera's optical axis), see Figure 23. Because the motion of the hexapod robot (especially

smoothness of the motion) is influenced by the robot's motion gait, experiments have been performed for two different gaits: the default tripod gait and the adaptive tripod gait. Experiments also contain trials captured while the robot traverses both the flat and uneven terrains, which was created from the corridor of wooden blocks covered by a carpet of the artificial grass. The artificial grass on the wooden blocks makes the surface of the terrain smoother; so, it improves traversability of the terrain by the default gait. On the other hand, soft terrains like artificial grass induce false surface detections, when the adaptive motion gait is used. All trials have been captured with the stereo camera Tara at the frame rate of 60 FPS.



(a) Flat terrain

(b) Uneven terrain

Figure 24: Absolute trajectory error

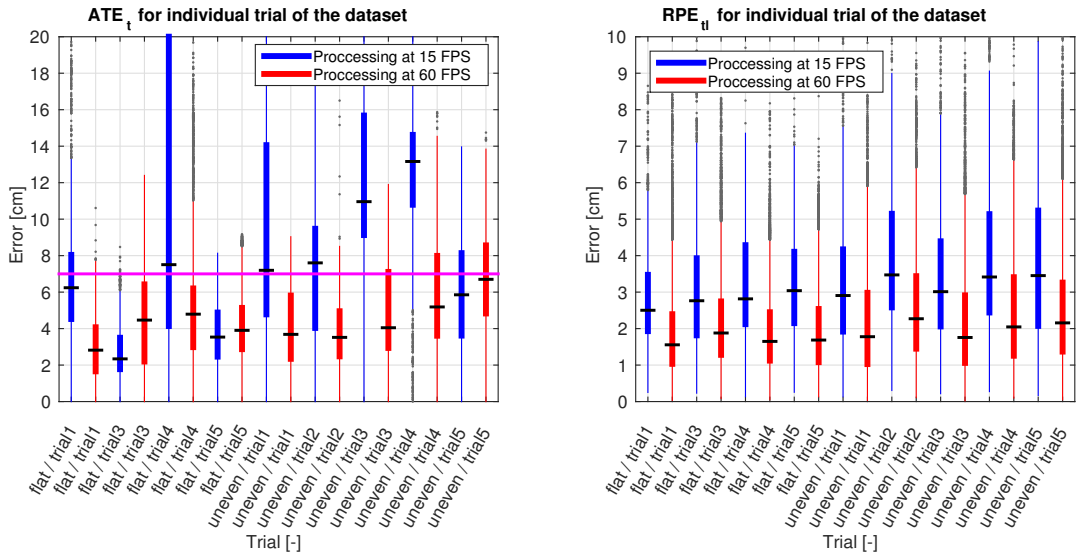
Results of the evaluation are shown in Figures 24a and 24b. Based on the results depicted in Figure 24a, it can be concluded that the absolute error of the trajectory estimate is lower when the robot walks on a flat terrain using the default motion gait except the case when the heading deflection (ψ) is 60 deg. Nevertheless, Figure 24b shows that when the robot walks on an uneven terrain it is better to use the adaptive gait because the error of the trajectory estimate is lower, except the case where $\psi = 60$ deg. These results are probably induced by the overall smoothness of the robot's motion. The hexapod crawling robot equipped with the adaptive motion gait moves at first with legs and than with the body of the robot. On the other hand hexapod robot equipped with default gait moves with the body almost continuously which makes the robot's motion smooth on a flat terrain, but on the uneven terrain many additional non-smooth motions are induced by slipping of the robot from obstacles.

5.3 Summary of experiments

In this section, the most important results of the performed evaluation are described.

5.3.1 Impact of higher framerate on localization precision

During the evaluation of the SLAM systems described in Sections 5.2.1 and 5.2.2, sensor data were processed at 15 FPS. The experimental results summarized in Tables 2 and 3 show that the localization with the smallest translation error was provided by stereo and RGB-D versions of ORB-SLAM. Nevertheless, in some trials, the localization provided by ORB-SLAM failed. These fails were caused by too fast changes of the camera field of view induced mostly by the high speed of rotation at the corners of the trajectory or by the uneven terrain.



(a) Absolute error of the trajectory (magenta line marks 7 cm)

(b) Relative error of the trajectory

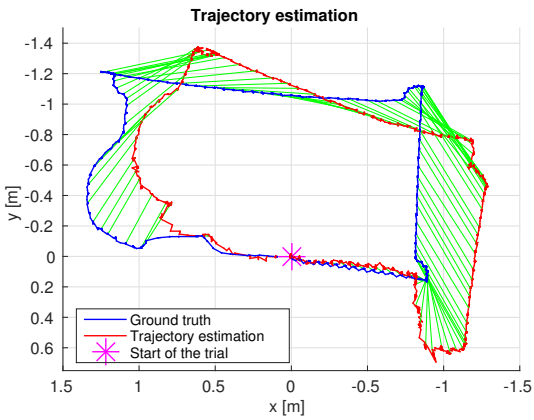
Figure 25: Impact of a higher frame rate on the accuracy of localization provided by ORB-SLAM. Trial 2 captured in the flat terrain was omitted in the comparison because at the frame rate of 15 FPS, the ORB-SLAM was unable to localize the robot.

The problem of too fast changes of the camera field of view can be solved by reduction of the robot's speed or by processing of the video at higher frame rate. The video stream provided by Tara in Setup 1 and Setup 2 was captured at frame rate of approximately 60 FPS, thus it was possible to evaluate an impact of the higher frame rate by processing using ORB-SLAM. At first, the trials where the tracking while processing at 16 FPS failed were processed at 60 FPS by ORB-SLAM. In all these trials, ORB-SLAM's tracking succeeded in the estimation of the trajectory. Afterwards, first five trials from both the flat and uneven terrain were processed at 60 FPS by ORB-SLAM at close loop scenario. Resulting trajectories were evaluated and the most important results are shown in Figures 25a and 25b.

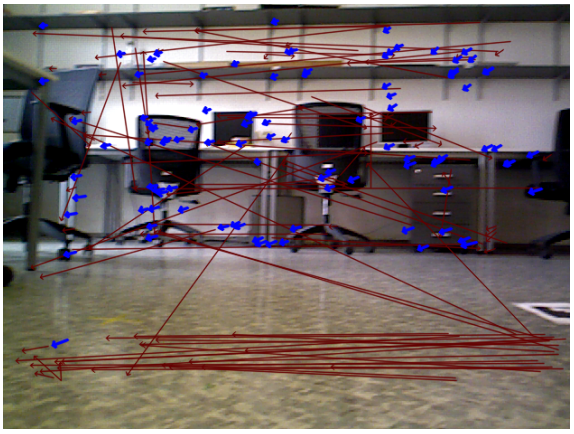
Figure 25b shows that the relative pose error of the trajectory estimates provided by ORB-SLAM at 60 FPS is lower at all evaluated trials than the error of the trajectory estimates provided by ORB-SLAM working at 15 FPS. Note, there are some trials where the absolute trajectory error was lower for method working at 15 FPS. However, it can be concluded that localization provided by the ORB-SLAM at 60 FPS is generally more accurate because the absolute pose error is in all cases lower than 7 cm; so, the estimation is more reliable.

5.3.2 High error of RGB-D SLAM's estimates

Based on Tables 2 and 3, it can be concluded that in both the Setup 1 (Section 5.2.1 and Setup 2 (Section 5.2.2) RGB-D SLAM generally provides trajectory estimates with higher error than the other SLAM systems. The problem affecting the accuracy of most trajectory estimations provided by RGB-D SLAM occurred when the robot turns at the first corner of a rectangular trajectory, see Figure 26a. This was probably caused by repeating feature pattern on the wall in front of the robot, which induced many false correspondences between images (depicted in Figure 26b), and therefore, the RGB-D SLAM calculated wrong geometrical transformations between the camera poses. Trajectory estimates provided by the ORB-SLAM and the S-PTAM were not affected by repeating of feature patterns because both algorithms use tracking of features.



(a) Trajectory estimated by RGB-D SLAM



(b) False correspondences (red arrows) in the critical section

Figure 26: High error of RGB-D SLAM's estimates

5.3.3 Consequences of S-PTAM Parametrization

S-PTAM provided trajectory estimations with generally second highest error in both the flat and uneven terrains. The main hypothesis, why the error was that high is that the used parametrization use a large neighborhood of tracked feature to find the correspondences in the following image. A large size of the neighborhood was chosen because when the robot traverses the uneven terrain, the distance between the corresponding features in sequential images is high.

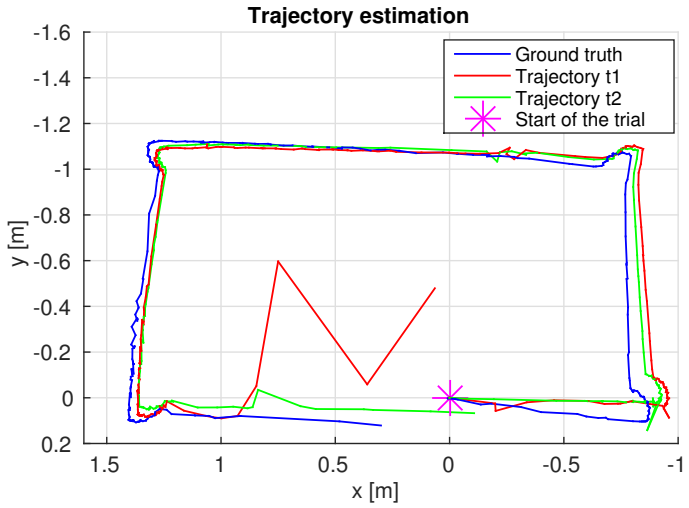


Figure 27: Comparison of trajectory estimations provided by S-PTAM with two different parametrizations. Trajectory $t1$ was estimated with parametrization using larger neighborhood and trajectory $t2$ was estimated using smaller neighborhood.

A special experiment was designed to verify this hypothesis. During this experiment, the trial captured on a flat terrain was processed with two S-PTAM parametrizations; the first parametrization $t1$ used a large neighborhood to find feature correspondences and the second one $t2$ used smaller neighborhood. The trajectory estimates provided by S-PTAM using both parametrizations have similar medians of ATE_t and RPE_{tl}. However, in Figure 27, it can be seen that the trajectory estimated by S-PTAM using parametrization $t1$ contains some outliers, which would make the localization unusable but estimation using parametrization $t2$ is free of such outliers.

5.3.4 Comparison of loop closure strategies

Based on the values of ATE_t, and ATE_ϕ presented in Tables 2 and 3, it can be concluded that the loop closures signifficantly improve the accuracy of localization provided by RGB-D SLAM even if the method faces problem that is described in Section 5.3.2. On the other hand, the same conclusion cannot be stated in the case of ORB-SLAM,

which provides very accurate localization even in large open loop scenarios.The only significant improvement on accuracy of the localization can be observed by comparing the end distances, which are generally lower for all methods that enables loop closing, see Tables 2, 3, and 4.

The noteworthy fact is that the stereo version of ORB-SLAM was unable to close the loop five times in Setups 1 and 2 together. However, the RGB-D version of ORB-SLAM as-well-as the RGB-D SLAM were able to close the loop in all trials in both setups. One of the reasons for this behavior is probably caused by a very high confidence needed for the loop closure, which prevents false loop closing. Nevertheless, the ORB-SLAM's module for relocalization is able to detect already seen scene from more difficult viewpoints.

Unlike the ORB SLAM, RGB-D SLAM is capable of detecting features that were already observed from relatively distant places and different directions. On the other hand, when RGB-D SLAM detects a loop closure, it starts the global optimization of the trajectory (see Section 3.2) and this procedure stops the whole SLAM system until it finishes the calculation. The time required for the calculation depends on a size of the pose graph, which has to be optimized, but even for small pose graphs used in experiments described in Section 5, the calculation takes approximately 2.5 s. It is possible to set how often RGB-D SLAM starts the global optimization, but for the experiments where the robot moves fast it could result in loss of localization.

5.3.5 Suggestions for the active localization

In this section, improvements of localization using evaluated SLAM systems based on principles of the active localization approach are suggested.

As it was shown in Section 5.3.3, the precision of the localization provided by S-PTAM depends on the size of the neighborhood, which is searched during the tracking of the image feature. A small size of the neighborhood prevents false matches of image features. The suggestion for this system is to adaptively change the size of the neighborhood based on the currently traversed terrain. Another way to improve the localization is to use a different version of S-PTAM, which predicts features positions in the new frames by fusing visual odometry with IMU measurements [23]. Prediction of the next feature positions enables to use a much smaller neighborhood than it was used during the experiments in this thesis.

Stereo and RGB-D versions of ORB-SLAM provided generally the most accurate localization of the hexapod walking robot, but the main problem was the reliability of the localization. Reliability of the localization provided by ORB-SLAM can be increased by processing video stream at higher frame rate as it was shown in Section 5.3.1. However, in some cases, limited computational resources does not allow this; so, the reliability can be also improved by slowing down the angular velocity of the robot.

In Section 5.2.3, it is shown, that the deployability of the monocular version of ORB-SLAM strongly depends on the turn radius. Thus, it is necessary to establish the

minimal turn radius of the robot. Another important thing is to make loop closures as often as possible because the scale of the map used by monocular version of ORB-SLAM may drift rapidly as it was shown in Figure 22b.

The disadvantage of RGB-D SLAM is very computational demanding process of the global map optimization, which takes very long time as it was mentioned in Section 5.3.4. During this computational period, the localization of the robot is not provided, which can result in fail of localization if there was significant change of the viewpoint. The fail of the localization can be avoided by measuring frequency of provided localization by RGB-D SLAM. When the frequency significantly drops down, than also the speed of the robot should by decreased to avoid localization failure.

General ideas based on active localization that can be use together with any of evaluated SLAM system are: 1) revisit places to close loops, 2) swich gaits based on the traversed terrain, and 3) control the gaze direction. The importance of the first idea was already discussed in Section 5.3.4, the second idea is based on evaluation in Setup 4. The last idea to control gaze direction is also based on evaluation in Setup 4, where Figures 24a and 24b show that when the heading of the robot is deflected by angle $\psi = 30$ deg, the trajectory error is lower than for $\psi = 0$ deg. This means that the robot should move with deflected heading to improve its localization, especially when it is possible to observe reliable image features.

Part 6
Conclusion

This thesis presents an experimental study of five SLAM systems (RGB-D SLAM, S-PTAM, and three versions of ORB-SLAM: stereo, monocular, and RGB-D) all deployed on the real hexapod walking robot in multiple different experimental scenarios. A thorough evaluation using proposed method has been performed on altogether 87 captured trials to compare the precision of aforementioned SLAM systems.

The results of the evaluation indicate that the localization with the highest precision was provided by stereo and RGB-D version of ORB-SLAM, the localization of the second highest precision was provided by S-PTAM. The localization with the worst precision was provided by RGB-D SLAM, which suffer from many false matches, thus wrong camera pose estimates.

Moreover, the evaluation reveals the individual factors influencing the precision of visual localization of the hexapod walking robot. Among the most prominent ones, it has been shown that tracking of image features can significantly increase reliability of feature matching, especially when the selected region for matching is small. Another important observation is that the monocular SLAM is not suitable for the localization in setups, where the robot turns with a small turn radius. This disadvantage can be avoided by enlarging the turn radius of the robot. The noteworthy result is also the fact that the deflection of the robot's heading can improve the localization. Based on the evaluation results, suggestions that follow active localization principles have been raised to improve the precision of hexapod robot's localization.

Therefore I am convinced that the main goals of this thesis were achieved. In future work, I would like to deploy SLAM system on hexapod walking robot utilizing the proposed improvements based on active localization to deal with the fully autonomous navigation in exploration scenarios.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT press, 2005.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a Versatile and Accurate Monocular SLAM System," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[3] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard, "An Evaluation of the RGB-D SLAM System," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 1691–1696.

[4] T. Pire, T. Fischer, J. Civera, P. De Cristóforis, and J. Jacobo Berlles, "Stereo Parallel Tracking and Mapping for Robot Localization," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 1373–1378.

[5] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 573–580.

[6] E. Olson, "AprilTag: A Robust and Flexible Visual Fiducial System," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2011, pp. 3400–3407.

[7] E. Lubbe, D. Withey, and K. R. Uren, "State Estimation for a Hexapod Robot," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 6286–6291.

[8] T. Pire, T. Fischer, and J. Faigl, "Impact Assessment of Image Feature Extraction on the Performance of SLAM Systems," *Acta Polytechnica CTU Proceedings*, vol. 2, no. 2, pp. 45–50, 2015.

[9] D. G. Lowe, "Distinctive Image Features from Scale-invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[10] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[11] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary Robust Independent Elementary Features," in *European Conference on Computer Vision (ECCV)*, 2010, pp. 778–792.

[12] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An Efficient Alternative to SIFT or SURF," in *International Conference on Computer Vision (ICCV)*, 2011, pp. 2564–2571.

[13] E. Rosten and T. Drummond, "Machine Learning for High-speed Corner Detection," in *European Conference on Computer Vision (ECCV)*, 2006, pp. 430–443.

[14] P. Rosin, "Measuring Corner Properties," in *Computer Vision and Image Understanding*, vol. 73, no. 2, 1999, pp. 291–307.

[15] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 834–849.

[16] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras," *arXiv preprint arXiv:1610.06475*, 2016.

[17] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle Adjustment – A Modern Synthesis," in *Lecture Notes in Computer Science*, vol. 1883, 2000, pp. 298–372.

[18] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D Mapping with an RGB-D Camera," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177–187, 2014.

[19] G. Bradski and A. Kaehler, *Computer Vision with the OpenCV Library*. O'Reilly Media, 2008.

[20] P. Čížek and J. Faigl, "On Localization and Mapping with RGB-D Sensor and Hexapod Walking Robot in Rough Terrains," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 2273–2278.

[21] M. Nowicki, D. Belter, A. Kostusiak, P. Čížek, J. Faigl, and P. Skrzypczynski, "An Experimental Study on Feature-based SLAM for Multi-legged Robots with RGB-D sensors," *Industrial Robot: An International Journal*, vol. 44, no. 4, pp. 320–328, 2017.

[22] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[23] T. Fischer, T. Pire, P. Čížek, P. De Cristóforis, and J. Faigl, "Stereo Vision-based Localization for Hexapod Walking Robots Operating in Rough Terrains," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 2492–2497.

[24] J. Shi and C. Tomasi, "Good Features to Track," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1994, pp. 593–600.

[25] W. Burgard, D. Fox, and S. Thrun, "Active Mobile Robot Localization," in *IEEE International Joint Conference on Artificial Intelligence (IJCAI)*, 1997, pp. 1346–1352.

[26] T. Vidal-Calleja, A. J. Davison, J. Andrade-Cetto, and D. W. Murray, "Active control for single camera SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2006, pp. 1930–1936.

[27] T. Vidal-Calleja, A. Sanfeliu, and J. Andrade-Cetto, "Action Selection for Single-Camera SLAM," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 6, pp. 1567–1581, 2010.

[28] J. Bayer, P. Čížek, and J. Faigl, "On Construction of a Reliable Ground Truth for Evaluation of Visual SLAM Algorithms," in *Acta Polytechnica CTU Proceedings*, vol. 6, 2016, pp. 1–5.

[29] T. Krajník, M. Nitsche, J. Faigl, P. Vaněk, M. Saska, L. Přeučil, T. Duckett, and M. Mejail, "A Practical Multirobot Localization System," *Journal of Intelligent and Robotic Systems*, vol. 76, no. 3-4, pp. 539–562, 2014.

[30] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an Open-source Robot Operating System," in *IEEE International Conference on Robotics and Automation (ICRA): Workshop on Open Source Software*, 2009.

[31] L. Černý, P. Čížek, and J. Faigl, " On Evaluation of Motion Gaits Energy Efficiency with a Hexapod Crawling Robot," in *Acta Polytechnica CTU Proceedings*, vol. 6, 2016, pp. 6–10.

[32] J. Mrva and J. Faigl, "Tactile Sensing with Servo Drives Feedback only for Blind Hexapod Walking Robot," in *10th International Workshop on Robot Motion and Control (RoMoCo)*, 2015, pp. 240–245.

[33] D. Belter, P. Łabecki, and P. Skrzypczyński, "Adaptive Motion Planning for Autonomous Rough Terrain Traversal with a Walking Robot," *Journal of Field Robotics*, vol. 33, no. 3, pp. 337–370, 2015.

[34] Collective of autors, "Tara - USB 3.0 Stereo Vision Camera," accessed May 20, 2017. [Online]. Available: https://www.e-consystems.com/3D-USB-stereo-camera.asp

[35] G. Klein and D. Murray, "Parallel Tracking and Mapping on a Camera Phone," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2009, pp. 83–86.

[36] S. Gauglitz, C. Sweeney, J. Ventura, M. Turk, and T. Höllerer, "Live Tracking and Mapping from Both General and Rotation-only Camera Motion," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2012, pp. 13–22.

[37] M. Menze and A. Geiger, "Object Scene Flow for Autonomous Vehicles," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3061–3070.

CD content

Table 5 lists names of all root directories on CD together with their content.

Directory name	Description
`/mt`	Bachelor thesis in pdf format.

Table 5: CD content.