Enabling the next generation of smarter self-navigating robots using Hybrid ToF

R. Leitner, E. Steurer, A. Berger, W. Bell

Edition: v1.0
Date: 04/2024

www.infineon.com/real3

Infineon

Table of contents

1   Abstract

Older robot vacuum cleaners used a random, inefficient, and slow cleaning pattern. Smart robots generate a map of their environment using sensors and simultaneous localization and mapping (SLAM) to localize, providing users with a floorplan from which areas can be selected for cleaning, or restricted from access. This whitepaper describes designing slimmer robots with new features using an approach with novel time-of-flight cameras.

Visual-SLAM (vSLAM) uses ≥1 cameras and requires high computational power to extract depth from the 2D images captured. SLAM uses depth-sensor time-of-flight (ToF) cameras providing true hi-res 3D images. Depth cameras have leaner and more efficient SLAM implementations compatible with embedded processing platforms for applications such as cleaning robots, warehouse robots, tracking drones, etc.

REAL3™ hybrid Time-of-Flight (hToF) combines a single ToF image sensor with two illumination types addressing:

– Provide precise long-distance 3D point-cloud data enabling SLAM
– Depth data with a high lateral resolution for advanced obstacle avoidance
– Accurate close-range data with high resolution to allow cliff detection

Infineon and partners developed hToF as a cost-efficient depth sensing technology for consumer products. Particularly for consumer robotics, smaller hToF sensors help design slimmer robots than laser-distance-sensor (LDS) tower robots. This whitepaper implements hToF sensors for consumer robots, presenting excellent map accuracies achieved with open-source SLAM implementations, as well as the reduced processing resources needed.

The unique hToF technology uses Infineon's high-resolution REAL3™ ToF sensor, a homogenous flood illumination, and a powerful spot grid illumination offering a cost-efficient solution. The hToF camera's broad, 110° horizontal field-of-view (FoV) is ideal for both SLAM and obstacle avoidance. hToF works in all lighting conditions (complete darkness to bright sunlight) across a large variety of furniture and floor reflectivity and textures. This is a huge advantage over structured light (SL) that struggles in bright sunlight, and also over stereo vision (SV) that struggles in darkness and with repetitive textures. Older robots were routinely immobilized under low-clearance furniture objects. Smart robots with hToF technology navigate better even under lower-clearance furniture, thanks to the reduced height and also the additional height-clearance information from the hToF sensor.

In an evaluation study, an R&D robot with two hToF cameras has been used in various office and home environments. The hToF sensor solution enables the robot to generate precise and consistent maps with excellent accuracy even for challenging environments with glass walls, non-rectangular rooms, dark floor material, and cluttered furniture objects. 3D depth image data by hToF handles glass walls properly, as the frame is detected as a wall.

The computational effort for hToF data's depth processing using pmd's library, and SLAM using the open-source Google Cartographer implementation, is benchmarked on three embedded SoC platforms used in robotics: NVIDIA Jetson Nano, Qualcomm RB5, and Raspberry Pi 3B/4B. We show that depth processing for hToF and corresponding SLAM algorithms can be executed on a single A53 (Raspberry 3B), in stark contrast to vSLAM systems that usually need ≥2 cores for their processing due to the high computational load of the required correspondence search.

In summary, the hToF technology with Infineon's REAL3™ sensors along with pmd's depth processing offer a powerful solution for consumer robots simultaneously supporting SLAM, advanced obstacle avoidance, and cliff detection. All this while reducing overall BOM cost, compute demand, and system complexity. The maps generated by open-source SLAM algorithms are accurate and reliable and prove the fidelity of the hToF spot-illuminated depth data. The solution is computationally lean, requiring only one core for depth processing and SLAM computation.

2   Introduction

Almost every household currently has one or more robot vacuum cleaners (RVC) and the trend of their usage is increasing. While the very early RVCs used a random pattern to clean a room, newer RVCs are much smarter and autonomously generate a map of the flat, house, or office they are operated in and use sensors to localize themselves during operation. This allows the user to select the room(s) to be cleaned in a smartphone app and the robot to find these rooms and the charging station on its own.

The task of a robot to generate a map of its environment and localize itself is usually attributed as SLAM or simultaneous localization and mapping. Since the publication of early SLAM approaches in a seminal article [1] in 1986, a plethora of different SLAM algorithms has been presented [2]–[7]. Among the multitude of SLAM approaches, the most important distinction is the difference between visual-SLAM (vSLAM) and classical point-based SLAM. While the first uses one or more monochrome or color camera sensors and requires a lot of computational power, the second uses depth sensors such as Time-of-Flight (ToF) camera sensors that provide implicit scene depth information without the need to establish correspondences based on the appearance of salient points.

Thus, point-based SLAM allows much leaner and computationally more efficient algorithms and is more suitable for embedded processing platforms usually used in cleaning robots. The novel Hybrid-ToF (hToF) technology goes even beyond this and combines a single ToF sensor and two types of illumination to address the requirements of SLAM, obstacle avoidance, and cliff detection at once, providing a powerful sensor solution for consumer robotics as shown in Figure 1.

Figure 1   Primary robotic use cases addressed by hToF: SLAM and navigation, advanced obstacle avoidance, and cliff detection

The advantage of SLAM based on hToF data is to smartly avoid areas occupied by static objects like chairs, tables, or other furniture that could have complex 3D shapes requiring special paths to ensure that the RVCs are neither bumping into them nor getting stuck. Sophisticated RVCs with hToF depth cameras can avoid moving objects with the advanced obstacle avoidance approach and also avoid obstacles with different clearance heights. They can elegantly handle furniture objects such as couches and cabinets that do not have the same clearance height by making decisions based on the camera's depth measurements. An object that has enough height clearance will be cleaned underneath while obstacles that are too low are avoided to prevent the RVC from getting stuck. Depending on the camera's resolution these RVCs can even classify objects and decide how to handle toys, plants, figures, clothes, or other obstacles by not even coming in contact with them.

Overall, SLAM features have improved the way in which cleaning robots help us in the everyday task of keeping our flats and houses clean. Depth cameras based on hToF technology and ToF imaging sensors provide reliable 3D data measuring the RVC's environment in every lighting condition from complete darkness to bright sunlight. This is the advantage that serves as the vital basis to achieve excellent results using efficient and powerful SLAM algorithms.

This whitepaper is focused on SLAM for consumer robotics, particularly RVCs. Additional whitepapers regarding advanced obstacle avoidance and cliff detection using hToF are planned in the future.

3   Background

Depth sensing technologies provide valuable additional information in various applications, such as augmented reality, face authentication, robotics, and autonomous vehicles. With depth data, these devices get access to better measurements for sampling the 3D environment, where persons and objects are, details about their movements, and so on. Obviously, distances cannot be measured directly without contact. Thus, various indirect techniques facilitating different sensors have been developed: time-of-flight, triangulation, stereo/multi-view, radar, lidar, stripe projection, ultrasound, AI-based monocular depth, etc. The scope of this whitepaper is robotics and thus we will briefly describe the relevant depth sensing techniques for robotics – time-of-flight, stereo vision, and structured light. Each technique has its unique set of advantages and drawbacks.

3.1   Time-of-Flight (ToF)

ToF cameras measure the time that it takes for a light signal emitted from a source (usually an infrared laser) to travel to the subject and back to the receiving sensor. Using the speed of light that is constant in the medium of the measurement, the distance can be accurately determined. Two principles have evolved for measuring the travel time accurately: direct (dToF) and indirect time-of-flight (iToF). dToF systems emit short laser pulses (usually single-digit nanoseconds) and use single-photon-sensitive detectors such as a SPAD (single photon avalanche diode) to generate a histogram of how many photons hit the sensor, and when. A peak detector determines the most probable distance to the object. The advantage of dToF cameras is that long distances up to 300 m can be detected, but the resource effort (time-to-digital converter, memory) on the chip detector is high. Thus, the number of pixels of dToF sensors is usually low.

In contrast, iToF cameras use a periodically modulated laser light (with a frequency between 20 MHz and 120 MHz) and measure the phase between the emitted signal and the received signal. Using the modulation frequency and the measured phases, the distance between the camera and the object can be determined precisely. The modulation frequency f limits the measurement range to the ambiguity range of (c/2)*f, where c is the speed of light. But this limit can be easily overcome by using two or more modulation frequencies. This resolves the ambiguity and extends the measurement range to the ambiguity range of the greatest common divider of the used modulation frequencies. An important advantage of iToF for consumer products is that the on-chip resource scaling is better and thus iToF imagers are small, cost-efficient, and available up to megapixel resolution.

3.2   Structured light (SL)

SL systems project a known pattern (often grids or stripes) onto a scene. A camera sensor captures the pattern's deformation created by the object's distance and shape, and the baseline between the projector and the camera. This disparity (pixel shift of the pattern) is used to calculate the depth information. The disadvantage of this method is the sensitivity of the approach to ambient light, object textures, and contrast edges. As the achievable accuracy is linearly dependent on the baseline between the sensor and the projector, the scope of miniaturization of the camera module without sacrificing accuracy requirements is usually limited.

3.3   Stereo vision (SV)

The term stereo vision originates from the fact that two cameras are used by SV systems with a baseline inspired by human binocular vision. Each camera captures an independent image and salient points are detected in both images. By establishing corresponding points in both images, the distance can be determined based on the disparity (usually, a horizontal pixel-shift in the camera images). The high computational effort for saliency detection and determining correct correspondences by checking the epipolar geometry predetermined during calibration is the main disadvantage of SV systems. SV systems are cost-efficient at the sensor-hardware BOM-cost level as only two cameras are needed. However, the high mechanical effort needed in maintaining a constant physical distance and the angle of both cameras over the product lifetime to guarantee that the factory calibration remains valid is another

disadvantage. SV systems require a sufficiently bright and homogenous ambient light. Active SV systems are used in low-light situations, resulting in a higher module cost due to additional illumination components.

3.4 REAL3™ hybrid Time-of-Flight solution (hToF)

hToF is a novel ToF paradigm developed by Infineon and its partners to enable a cost-efficient depth-sensing technology for consumer electronics. hToF combines two different illumination types with a single iToF imaging sensor. Particularly in robotics, the main application use cases of SLAM, obstacle avoidance, and cliff detection have contradicting requirements. While SLAM requires a reliable long-distance information but only low lateral resolution, obstacle avoidance and cliff detection need only close-distance information but a high lateral resolution.
hToF solves this dilemma by using a spot grid illumination for SLAM and flood illumination for obstacle avoidance and cliff detection as shown in Figure 2. The high optical power in each spot allows reliable long-distance depth data for SLAM, while the homogenous bat-winged flood illumination compensates the relative illumination fall-off of the lens towards the borders of the field of view (FoV). This provides a high lateral resolution based on the native iToF sensor pixel number. Due to the wide horizontal FoV of the hToF module, the solution becomes cost-efficient for consumer products.

Figure 2 **Overview of a hybrid Time-of-Flight module's characteristics, the spot and flood types of illumination, and its advantages for robotics that support SLAM, advanced obstacle avoidance, and cliff detection**

In the past, laser-distance-sensor (LDS) modules have been used to provide the required depth data for SLAM in consumer robotics. The disadvantage of LDS systems is that they are large, expensive, and comprise a motor and other mechanically moving parts such as a mirror, lowering lifetime and increasing maintenance costs. Additionally, the LDS tower increases the height of the robot. Since a hToF module is smaller, it can be integrated into the robot and thus allows building slimmer robotic devices. Despite the small size, hToF addresses all three robotic use cases due to the unique combination of a homogenous flood illumination with a powerful spot grid illumination and a high-resolution ToF sensor. This provides a cost-efficient solution and reliable depth data for SLAM algorithms while still offering a high lateral resolution for obstacle avoidance and cliff detection.

4 Robotic platform

The robotic R&D platform used for this evaluation study is based on the well-known TurtleBot3 platform, which has a large user-base in R&D, applied research, and education [4]. We extended the system by means of mounting two hToF modules on the top plate by utilizing 3D printed camera holders as shown in Figure 3 (left). We designed a custom holder for the OpenCR board of the TurtleBot3 platform that is responsible for motor control, IMU (inertial measurement unit), odometry handling, as well as providing the main power supply to the system. This custom holder ensures that the IMU is placed exactly at the turning center of the robot, improving SLAM results. Additionally, the OpenCR board is mounted using rubber elements implementing an efficient dampening to reduce the apparent mechanical vibrations of the robot reaching the IMU. Figure 3 (right) shows the Infineon Mobile Robot (IMR), which is another robotic R&D platform comprising Infineon power electronics and various sensors, such as two hToF camera modules for SLAM, obstacle avoidance, and cliff detection [5].
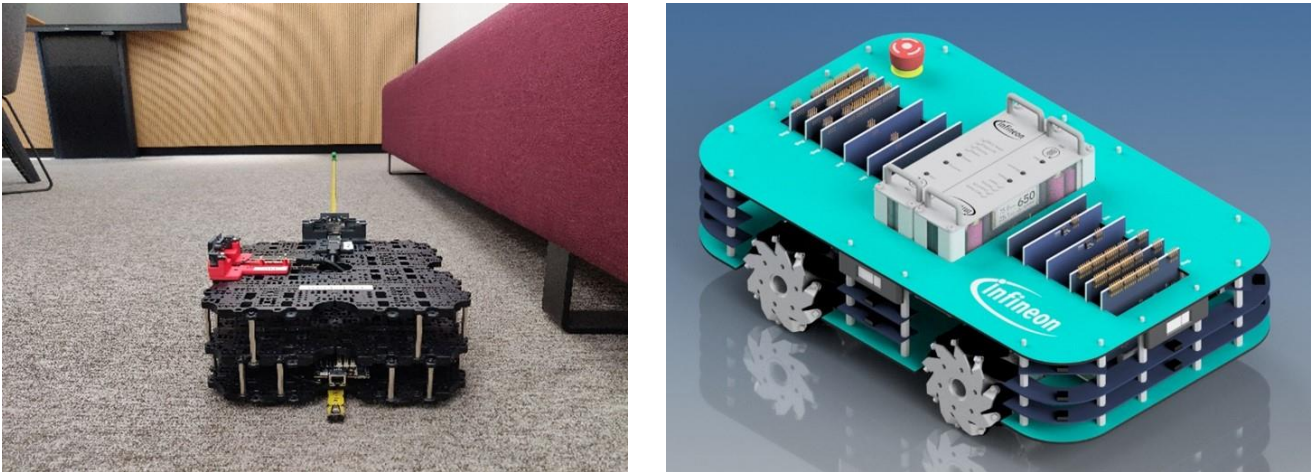

Figure 3 Robotic R&D platform TurtleBot3 with two hToF camera modules (left) and the Infineon Mobile Robot (IMR) with omnidirectional drive and also with two hToF camera modules

4.1 Components

The next sections give an overview and describe briefly each of the components in the R&D robot platform that are used to provide the excellent SLAM results achieved with hToF sensors. Via the URDF configuration file, Cartographer, the SLAM implementation used here, is provided with the position and orientation of the hToF cameras relative to each other as well as the robot center (`base_link` corresponding to the turning point of the robot in the middle of the front axle). Additionally, the position and orientation of the OpenCR board, precisely the IMU, is defined in the URDF file as well.

4.1.1 Qualcomm RB5

RB5 is a high-end embedded platform developed by Qualcomm targeting robotic applications. It houses the well-known mobile SM8250 chipset on the QRB5165 SoM incorporating four Arm® Cortex® A55 and four A77 processor cores, 8 GB LPDDR4, and a 128 GB Universal Flash Storage (UFS). It supports the ROS1 stack [6], which enables fast prototyping, integration, and evaluation of sensors in a robot system. Furthermore, such a platform provides enough memory for capturing large datasets (complete floor of a large office building) comprising several sensor data streams and raw uncompressed video data. However, it is important to note that it is not required to use such a high-end system to operate hToF, its depth processing, and a SLAM implementation. As described in detail in the benchmark results (section 7), a Raspberry 3B is sufficient and only a single A53 core is needed for depth processing and SLAM computation. Of course, the hToF depth processing and SLAM implementation run easily on a little core (A55) or a big core (A72) of the RB5 platform leaving enough headspace for additional R&D nodes, evaluation modules, etc. But typically, in the final consumer product the ROS environment is replaced by a bare metal or embedded Linux environment, reducing the compute demand even further.

4.1.2   OpenCR board

The OpenCR board is used as-received with the TurtleBot3 without any changes. It is a microcontroller board responsible for motor-control, IMU, odometry handling, as well as for supplying the correct voltages to SoC boards like RB5 or Raspberry. The hardware and firmware are both open-source and can be accessed from the corresponding GitHub repository [7].

4.1.3   hToF camera module

There are two hToF camera modules mounted on the system. The image sensor provides spot and flood iToF phase and pixel data over a MIPI CSI-2 interface (two lanes with 1 GB/s each). The Infineon MIPI2USB bridge CX-3 [8] receives the data via MIPI and forwards it over a USB 3.0 connection to the RB5 host system. pmd's depth processing library Royale is responsible for camera control, loading of sensor settings, as well as adjusting the exposure time during run-time based on the current scene. The camera including the additional CX-3 PCB is mounted on a custom designed 3D print on the top plate of the TurtleBot3 Waffle platform.

The details regarding the hToF camera module are available on the Infineon webpage [9]. Details about Infineon's REAL3™ iToF imager are available on Infineon's product webpage [10] and also on the webpage of our module maker partner, OMS [11].

Figure 4   hToF camera module with MIPI CSI-2 interface (left). An interface module comprising a MIPI2USB bridge based on Infineon's EZ-USB™ CX3 is available as well (right). This module is used in this evaluation study to connect the hToF camera modules to the RB5 platform.

4.2   System interconnects

The various components inside the R&D system are connected via USB connections. Both hToF MIPI camera modules are connected to our MIPI2USB bridge, which is then connected via USB 3.0 to the RB5 board. While the MIPI hToF modules are used in the end in consumer products, for R&D and evaluation, the USB hToF modules are more efficient by moving the MIPI integration to later project phases. The OpenCR board uses a USB 2.0 connection to communicate with the RB5. Figure 5 below shows the interconnection of the system.
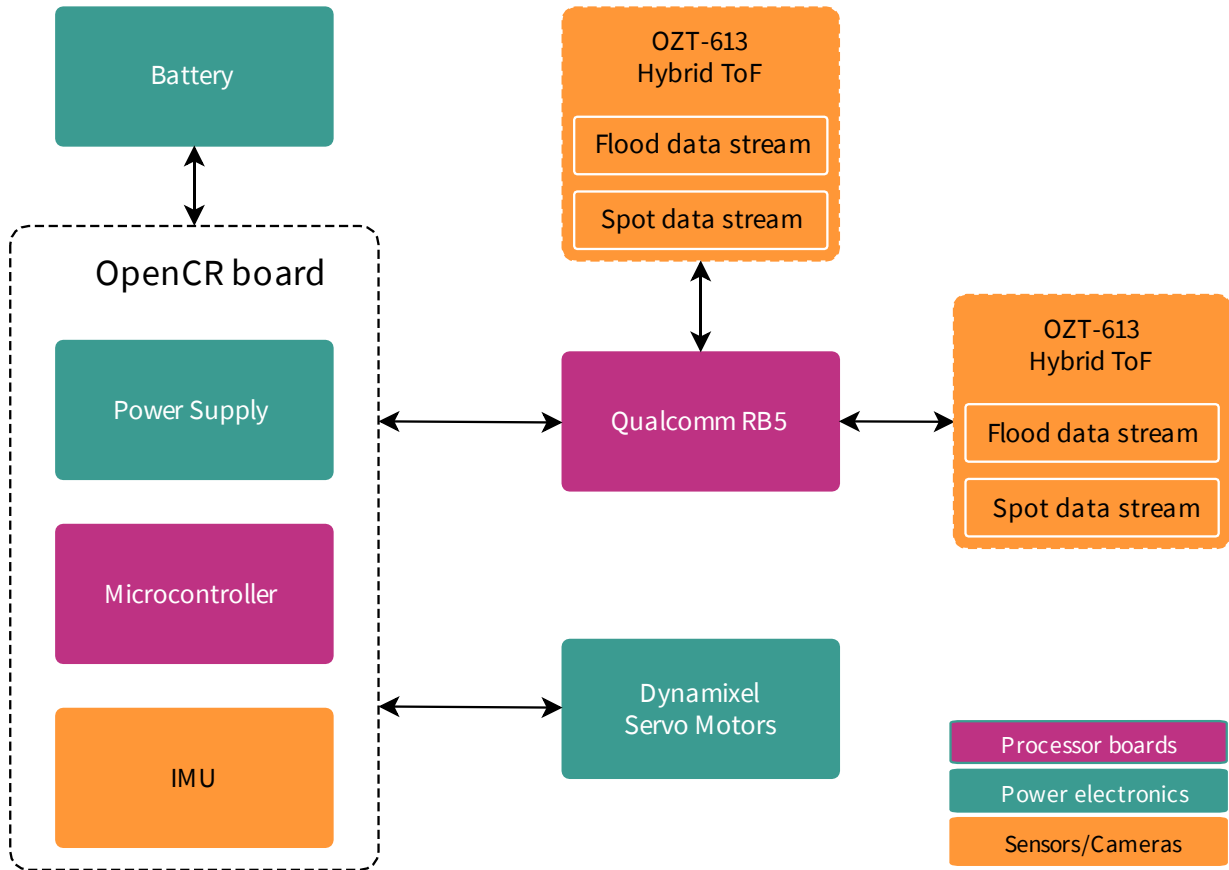

Figure 5   Interconnects of the main hardware components inside the R&D platform

4.3 Power supply

The main power supply is delivered via a 11.1 V/4000 mAh Li-Po battery attached to the OpenCR board. It is further distributed to the RB5 board by a regulated 12 V supply and a buck-boost converter with a maximum current rating of 5 A and 5 V. Therefore, it is possible to power the RB5 board directly, which subsequently powers the USB hToF modules. Additionally, power is delivered to the motors and the RC receiver. Using the described rechargeable battery, the R&D platform can be operated for several hours to acquire high-quality data in several SLAM recording sessions.

5   Software

5.1 Google Cartographer SLAM

The evaluation study has been conducted using the Google's Cartographer SLAM implementation [12] as it has been used in R&D across different hardware and software configurations. This approach has been proven to work in various environments, indoors and outdoors. It is a graph-based SLAM [12] which is split-up functionality-wise in local and global SLAM. Figure 6 shows the various components in Cartographer involved in establishing location and mapping functionality.
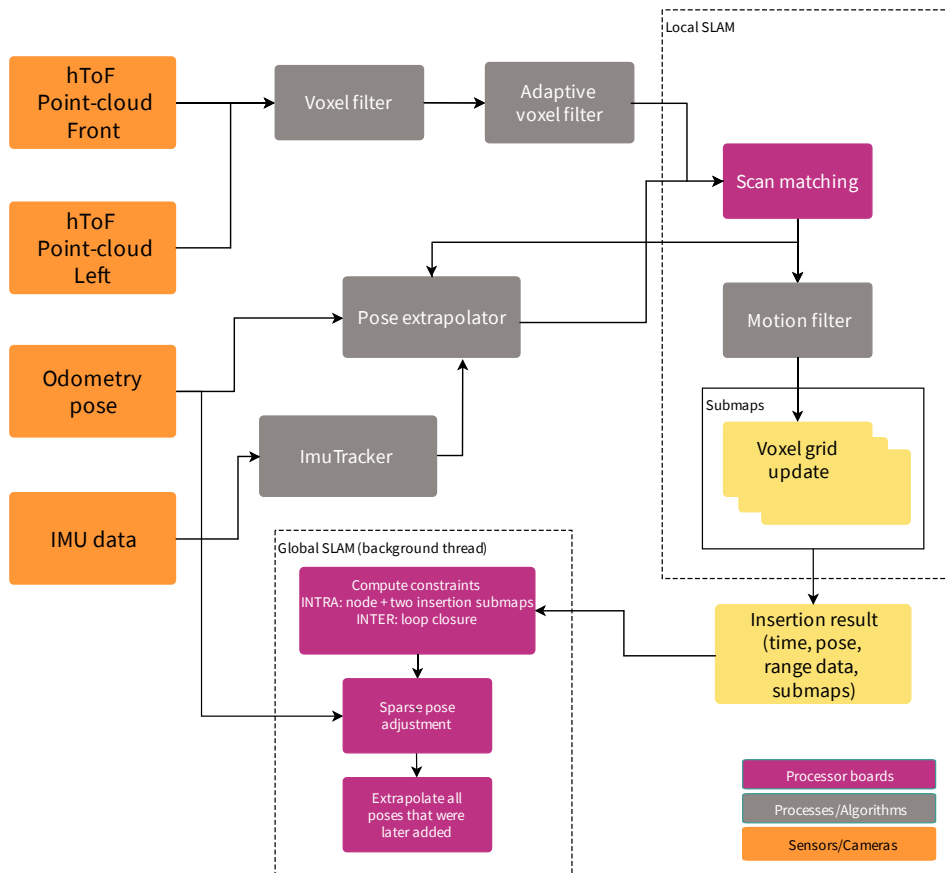

Figure 6   Overview of the Cartographer SLAM system (adapted from Google Cartographer documentation [12])

Supplying time-stamped sensor data is vital to the correct functionality of the algorithm. The easiest way to do this is using the `cartographer_ros` node that utilizes ROS to collect the data, provide an interface to the current pose of the robot, and also to the map that is continuously built. Two hToF point-clouds, an odometry pose, and IMU information are provided to Cartographer. The next sections give a high-level overview of local and global SLAM in Cartographer. The Cartographer docs page [13] is especially helpful when tuning Cartographer for a target platform and gives a more in-depth view on the topic. Furthermore, the GitHub page of Cartographer can be accessed to directly look at the implementation and all the configuration parameters available [14].

5.1.1   Local SLAM

The primary aim of local SLAM is to perform mapping and provide the robot with its immediate location within its surroundings. It has the requirement of working in soft real-time and is therefore optimized for speed and responsiveness. This is achieved by throwing away almost all sensor data initially prior to running the CERES-based scan matcher. The point-clouds are subsampled in cubes of constant size and only the centroid of the cube is kept after filtering. After a fixed voxel filter is applied, an adaptive voxel filter tries to maintain a certain target number of points.

After all sensor information is received and point-clouds are filtered, the CERES scan-matcher tries to align the filtered point-cloud data of the hToF camera modules with the current submap grid, starting with an initial pose guess taken from the pose extrapolator. The pose extrapolator itself calculates the pose-estimate based on odometry and IMU data and feedback of the scan matcher. For this to work accurately and provide good results, it is necessary to have the position of each component aligned correctly and represented in the unified robotics description format (URDF). Furthermore, it is required to find good values for the weights (`occupied_space_weight, translation_weight, rotation_weight`) that Cartographer uses when performing scan-matching. These weights are a dimensionless measure of trust in the involved sensor sources.

5.1.2   Global SLAM

During local SLAM the position error accumulates as time progresses. The task of global SLAM is to counteract this by re-adjusting the submaps to each other to create a consistent global map. This is done by optimizing the trajectory built by local SLAM. This optimization problem works on previously computed constraints, which are estimated poses of nodes (laser-scans) relative to a submap. They are further categorized into non-global (intra) and global (inter) constraints. Nonglobal constraints are constraints that belong to the currently built submap while global constraints are poses of nodes that are not part of the currently built submap, but close enough to be considered and a strong fit. The former is used for keeping the local trajectory coherent while the latter is responsible for loop closures. Cartographer does not build constraints for all nodes that are accumulated during run-time, but only samples them with a certain configurable sampling ratio. Sampling too many nodes slows down the global SLAM and prevents the detection of timely loop closures while sampling fewer nodes results in a bad loop closure result.

5.2 Depth processing

The hToF camera module delivers high-quality phase and grayscale images for each modulation frequency and illumination type including the HDR modes with short and long exposure times. These raw phase images are received by the depth processing library from pmd [15] to generate low-noise but high-resolution depth and amplitude images. For the results presented in this white paper, we used the depth processing of Royale SDK 5 and the included ROS1 camera nodes.

The advantage of Royale is that it includes libraries for both Windows and Linux that are optimized for Intel, AMD, and Arm® processors. Additionally, ROS1 nodes are available allowing direct and straight-forward integration into ROS1 systems (ROS2 nodes are available as well).

The depth processing used by Royale has an optimized processing pipeline for:
– Infineon-pmd 3D Time-of-Flight (ToF) cameras comprising flood and spot processing
– Lens and temperature correction
– Several lateral and temporal filtering options
– Multi-path interference suppression and output of depth map
– 3D point-cloud
– Amplitude
– Confidence images

5.3 ROS nodes

As previously described, the SLAM functionality is achieved by utilizing ROS. Each data source is a ROS node providing data to the `cartographer_ros` node that itself acts as an interface to the Cartographer library developed by Google. The Cartographer node itself publishes the current submaps, trajectories, and scans the matched points. The occupancy grid can then be conveniently viewed in `rviz` during the SLAM run. Figure 7 shows the ROS nodes and the flow of topics during the Cartographer runs.
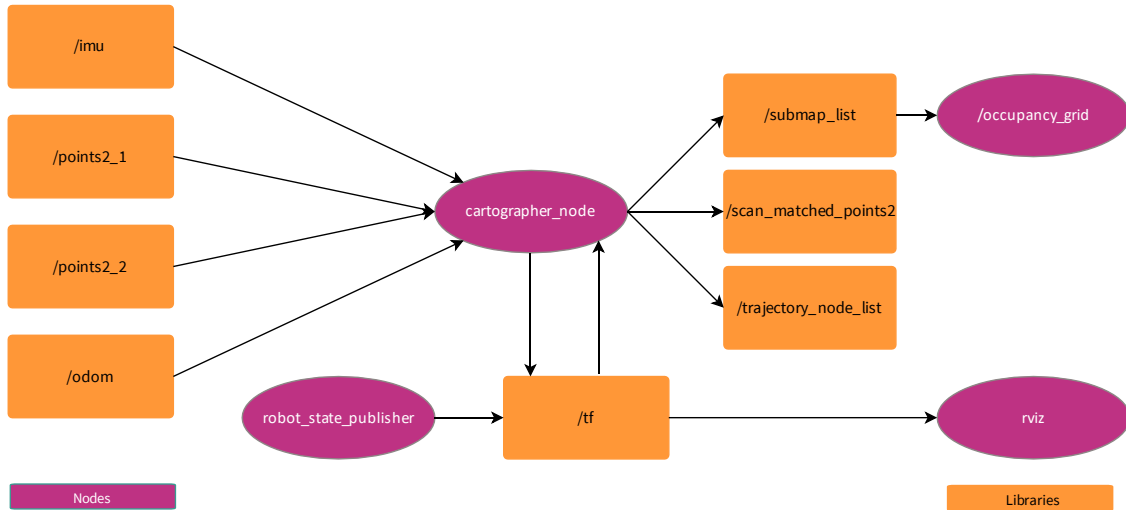

Figure 7   ROS nodes during the SLAM run

5.3.1   IMU

IMU data is supplied to the SLAM system by utilizing the IMU mounted in the center of the OpenCR board. This message contains the raw values of the gyro and the linear accelerations for x, y, and z. Additionally, a pose is calculated using a Madgwick filter [16], which is implemented in the firmware and fuses both data sources to obtain a pose estimation. This pose is also part of the IMU message provided to Cartographer.

5.3.2   Odometry

The odometry message provides Cartographer with a position calculated by the wheel encoders and an orientation based on the output of the Madgwick filter. Odometry has the tendency to drift over time that is compensated by the SLAM during run-time.

5.3.3   Point-cloud

The point-cloud data is provided by the two hToF cameras present in the system. The depth processing library (section 5.2) is used to control the cameras and receive the phase and pixel data of the two streams (spot and flood data) of each camera during run-time. Subsequent to the depth processing, the depth data is converted into the ROS-compatible PointCloud2 format. This way, Cartographer is agnostic to the data source that provides the depth data.

6 Map accuracy benchmarks

6.1 Local SLAM accuracy

Before assessing the global SLAM map accuracy, we optimized the local SLAM accuracy ensuring that the scan matching provides optimal results. Depending on the intended application scope, room size, and map extent vs. distance range of the used depth sensors, specific trade-offs have to be chosen to guarantee precise maps. For this, optimized trajectories served as ground truth and were compared with the poses generated by the local SLAM. For scenarios with larger rooms, we found out that keeping the maximum sensor distance range achieves better results, while for environments with smaller rooms limiting the distance range provided a better map accuracy overall. In the end, this parameter optimization has to be done for the final robot system, sensor configuration, and SLAM application scenario.

6.2 Global SLAM accuracy

Various maps have been generated with the robot driving through office floors and four-person flats in different sensor configurations. Generally, a robot equipped with one hToF camera already provides very good maps matching the ground truth of the floor plan. However, the larger FoV with two hToF cameras in the robot resulted in a higher map accuracy in less time as a single round in every room was sufficient. We compared the hToF-based SLAM-generated maps using the occupancy grid with 5 cm per pixel resolution provided by the SLAM implementation. This map is arbitrarily rotated depending on the robots starting direction. Thus, we aligned this map manually with the ground truth from the building's floor plan and refined the alignment transformation using an approach based on the normal distribution transform (NDT) [17]. To obtain the average map accuracy, point-clouds were computed from the SLAM-generated map and the ground truth, and the histogram of all the nearest point-to-point distance between both maps were determined. As for the average map accuracy, we used the statistical mean of all the point-to-point distances.

6.2.1 Office building floor with an area of 50 × 45 m²

We used the robot to conduct a map generation run driving through each room in an office building floor in our Infineon design center in Graz within an area of 50 × 45 m². In each room, the robot drove one round along the outer wall. The data acquisition took about 30 minutes due to the limited speed of the robot. Figure 8 shows the overlay of the map results of this run through all the office rooms and a histogram of all point-to-point distances. The average map accuracy of the generated map was 11.8 cm. Optimization runs with different weight parameters between hToF point-cloud data, IMU, and odometry unveiled that reducing the weights in global SLAM optimization for IMU and odometry and increasing the weight of the hToF point-cloud data improved the overall map accuracy.
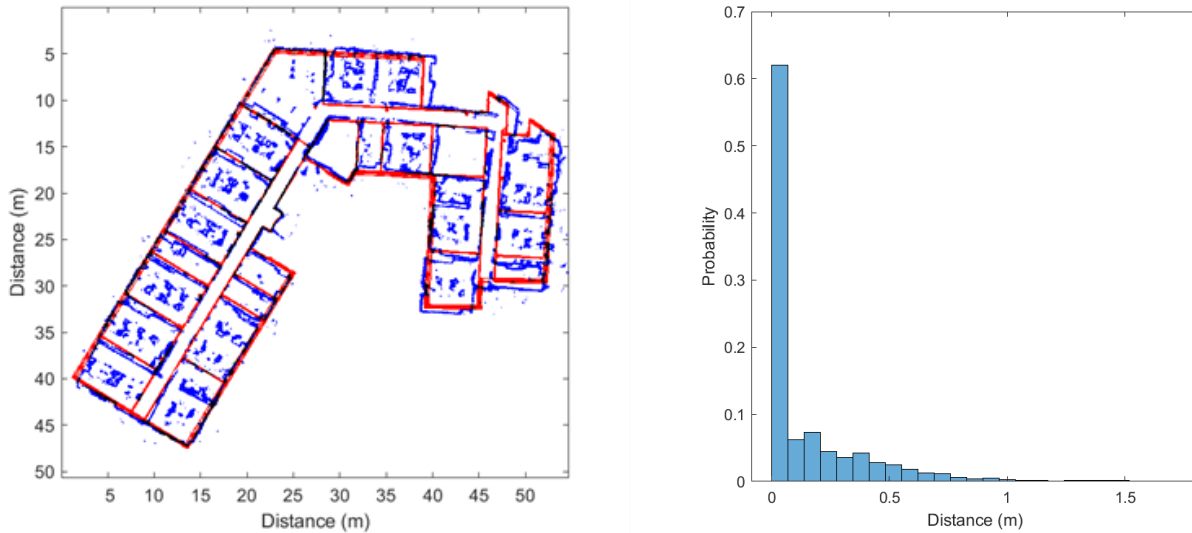

Figure 8 Map overlay of an office building floor with 50 × 45 m² with red pixels representing the ground truth based on the floor plan, blue pixels the SLAM-generated map, and black pixels if set on both maps (left). The distance histogram of each point-to-point distance showing an average map accuracy of 11.8 cm (right).

6.2.2 Map of a four-person flat with an area of 11 × 11 m²

As a second test set, we used the robot to drive through each room of a four-person flat with an area of 11 × 11 m². The data acquisition took about 8 minutes in this case. Figure 9 shows the overlay of the map results of this run and the histogram of all point-to-point distances. The average map accuracy of this test was 3.6 cm.
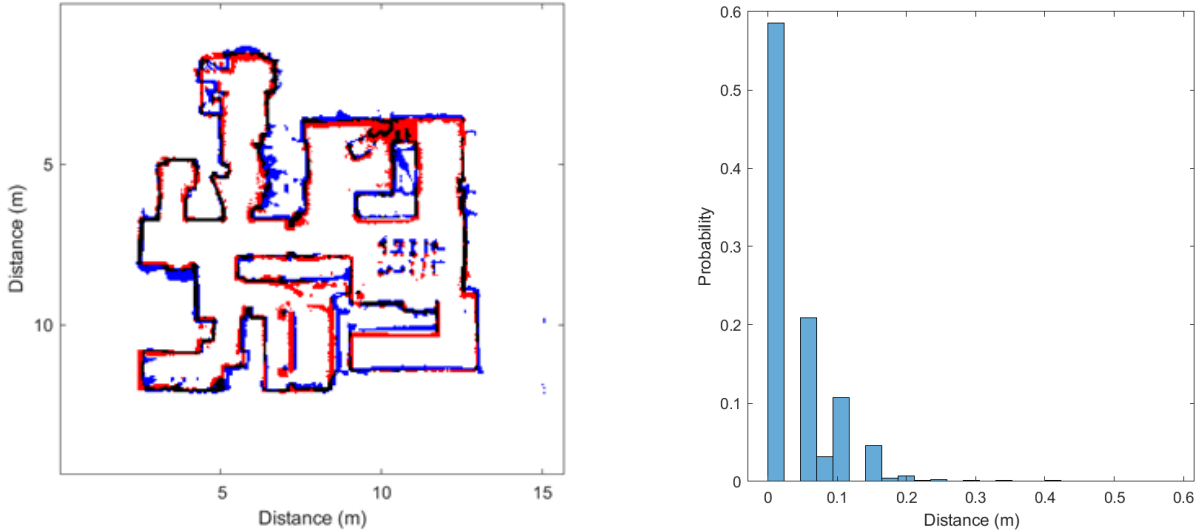

Figure 9 Map overlay of a four-person flat with 11 × 11 m² area with red pixels representing the ground truth based on the floor plan, blue pixels the SLAM-generated map, and black pixels if set on both maps (left). The distance histogram of each point-to-point distance showing an average map accuracy of 3.6 cm (right).

6.3 Localization accuracy

To assess the localization accuracy of SLAM implementations in robotics, they are compared to the ground truth provided by a high-accuracy system. Since these systems are expensive and not easily available, a simpler way of testing was conducted. We limited the dimensions to driving along a straight line with the robot stopping at random points and taking measurements of the position using a simple tape measurement. Table 1 shows the results of the Cartographer trajectory compared to the ground truth data taken from the measuring tape. The localization error is within ±1.5 cm and confirms the superior performance of a SLAM implementation using Infineon's hToF sensor solution.

Table 1 Ground truth measurement compared to SLAM localization

Distance (cm)	Localization error [cm]
22	0.23
42	-0.26
65	-1.39
111	-0.6
160	-0.83
186	-0.33
124	-0.89
91	0.76
27	0.62
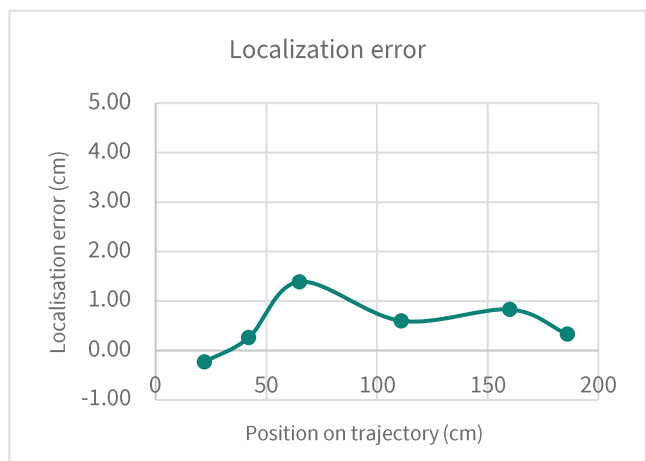
7   Processing benchmarks

In addition to excellent map accuracy results, the requirements regarding compute resources are key in any consumer product due to server-cost constraints. We used the ROS environment on Ubuntu and set the processor affinity in a way that the depth processing and the Cartographer threads are executed on a selected single core. We also ensured that the OS and ROS overhead is covered by the other cores. Then, we acquired the execution times per frame and processor load statistic when processing a pre-recorded RRF-file with the playback feature of the Royale SDK. The process/thread data available via "`/proc/<PID>/stat`" for the relevant processes `<PID>`, for the depth processing and the SLAM implementation were analyzed during run-time and averaged over the pre-recorded file comprising 600 hToF super-frames.

The following SoC platforms have been evaluated:

– NVIDIA Jetson Nano (Arm® Cortex® A57)
– Qualcomm RB5 (Arm® Cortex® A77 and A55)
– Raspberry Pi 3B (Arm® Cortex® A53) and 4B (Arm® Cortex® A72)

Two hToF use cases have been tested covering the relevant applications in consumer robotics:

– **Standard hToF:** Two modulation frequencies with four phase images each for flood and two modulation frequencies with four phase images each for spot. Each super-frame was acquired with 20 fps leading to 10 fps for flood and 10 fps for spot depth data simultaneously.

– **HDR hToF**: Two modulation frequencies with four phase images each for flood and two modulation frequencies with four phase images each for spot including two exposure times (short and long). Each super-frame was acquired with 20 fps leading to 10 fps for flood and 10 fps for HDR spot depth data simultaneously.

The execution times for the three evaluated SoC platforms and both hToF use cases, standard hToF and HDR hToF, are illustrated in Figure 10. Even on the weakest platform, the Raspberry 3B (Arm® Cortex® A53) the hToF processing stays below 20 ms execution time for standard hToF and 25 ms for HDR hToF.
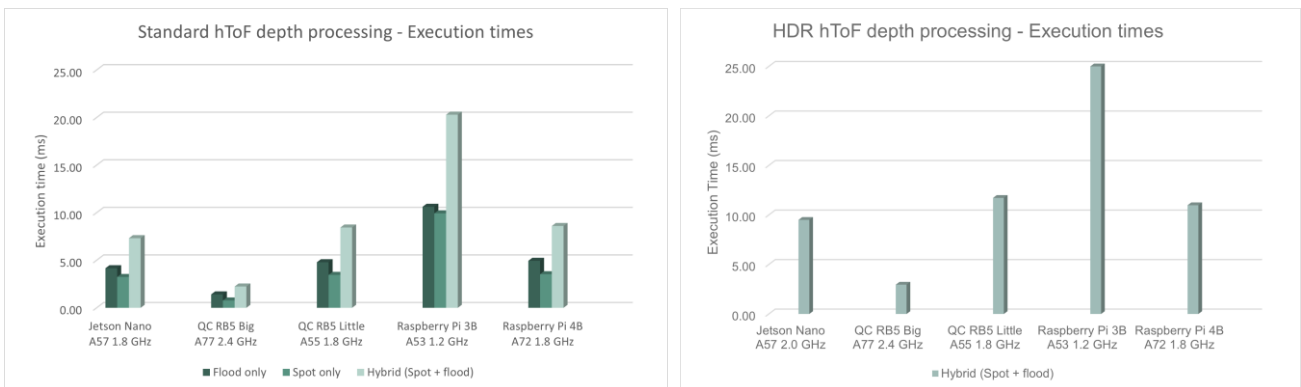

Figure 10 Execution times of standard hToF depth processing (left) and HDR hToF depth processing (right)

Additionally, the processor load has been assessed on the selected SoC platforms. In this case, the load for the corresponding SLAM algorithm (Google Cartographer) has also been measured. Figure 11 depicts the processor loads for the three SoC platforms and both hToF use cases. Combining the load paretos for hToF depth processing and SLAM leads to the conclusion that on each SoC platform the necessary computation can be executed on a single core, even on the smallest Raspberry Pi 3B Arm® Cortex® A53 at 1.2 GHz. This effectively demonstrates the efficiency of the hToF-based SLAM solution for embedded platforms.

Figure 11 Processor load for standard hToF depth processing (left) and HDR hToF depth processing (right)

8   Conclusion

The presented results effectively prove that the REAL3™ hybrid Time-of-Flight solution is the best sensor technology for consumer robotics to solve the three important application use cases – SLAM, obstacle avoidance, and cliff detection. The seemingly contradictory requirements for high lateral resolution close-range depth data for obstacle avoidance and cliff detection versus sparse long-distance point-cloud necessary for SLAM, can be met simultaneously with hybrid ToF camera modules that combine two illumination types, namely flood and spot, within one module.

Altogether, hToF represents a cost-efficient solution addressing consumer robotics replacing LDS towers and various other sensors allowing to design slimmer robots with more space for cleaning systems and the battery, while still providing additional data such as height clearance information. While a good map accuracy is achieved using a single hToF module, a better map accuracy is provided by robots with two hToF modules. Also, a faster map generation trajectory was made possible thanks to the larger FoV obtained with two hToF modules. Depending on the requirements for map accuracy, map generation time, and cost constraints the optimal hToF sensor configuration can be selected.

The excellent map accuracy with our demonstrator system based on the widely used TurtleBot3 in R&D shows that hybrid ToF provides high-quality depth data for SLAM using spot illumination. For very large maps such as an entire office building floor with an area of 50×45 m^2 the average accuracy is 11.8 cm, while for a four-person flat with an area of 11×11 m^2 an awesome map accuracy of 3.9 cm has been achieved. Comparing this with the 5 cm map (pixel) resolution used by the SLAM algorithm convincingly demonstrated the fidelity of the hToF-based SLAM approach for consumer robotics.

Despite all this, the hToF-based SLAM approach is lean on compute resources. We have effectively shown that all the processing can be executed on a single core, even a Raspberry 3B Arm® Cortex® A53. This includes the depth processing for two hToF cameras in HDR hybrid mode and the SLAM algorithm (without any code optimizations on the "original" Google Cartographer, only parameter optimization). This is a tremendous advantage for consumer robots as the processor load is reduced, thus freeing up compute resources for other application features.

In summary, Infineon's REAL3™ hybrid Time-of-Flight sensors along with pmd's depth processing offer a powerful solution for consumer robots simultaneously supporting SLAM, advanced obstacle avoidance, and cliff detection. The maps generated by open-source SLAM algorithms based on hToF spot-illuminated depth data are accurate and reliable. The solution is computationally lean, requiring only one core for depth processing and SLAM computation.

References

[1] R. C. Smith, and P. Cheeseman, "On the representation and estimation of spatial uncertainty," *International Journal Robotics Research,* p. 22, 1986.

[2] M. Dissanayake, P. Newman, S. Clark, and H. Durrant-Whyte, "A Solution to the Simultaneous Localisation and Map Building(SLAM) Problem," *IEEE Transactions on Robotics and Automation,* p. 14, 2006.

[3] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping problem," *18th national conference on Artificial Intelligence,* p. 6, 2002.

[4] Amsters R. and Slaets P., "Turtlebot3 as a Robotics Education Platform," p. 11, 2019.

[5] Infineon Mobile Robot IMR, [Online]. Available: https://www.infineon.com/cms/en/applications/robotics/development-platform/.

[6] Robot Operating System (ROS) webpage, [Online]. Available: https://www.ros.org/.

[7] OpenCR webpage, [Online]. Available: https://github.com/ROBOTIS-GIT/OpenCR.

[8] Infineon, EZ-USB™ CX3 MIPI CSI2 to USB 5 Gbps Camera Controller, Infineon Technologies AG, March 2024. [Online]. Available: https://www.infineon.com/cms/en/product/universal-serial-bus/usb-peripheral-controllers-for-superspeed/ez-usb-cx3-mipi-csi2-to-usb-5gbps-camera-controller/.

[9] hToF section on Infineon webpage, [Online]. Available: https://www.infineon.com/cms/en/product/sensor/tof-3d-image-sensors/tof-3d-image-sensors-for-consumer/#!highlights.

[10] Infineon REAL3 product webpage, [Online]. Available: https://www.infineon.com/cms/en/product/sensor/tof-3d-image-sensors/?redirId=197393.

[11] OMS webpage, [Online]. Available: http://www.o-film.com/en/field_inner_31.html#productsinner.

[12] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," p. 8, 2016.

[13] Google Cartographer Documentation, [Online]. Available: https://google-cartographer-ros.readthedocs.io/en/latest/algo_walkthrough.html.

[14] Google cartographer GitHub repository, [Online]. Available: https://github.com/cartographer-project/cartographer.

[15] pmd Royale SDK, [Online]. Available: https://3d.pmdtec.com/en/royale-software/.

[16] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm".

[17] P. Biber and W. Strasser, "The Normal Distributions Transform: A New Approach to Laser Scan Matching," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* Las Vegas, NV. Vol. 3, pp. 2743–2748, 2003.

[18] J. J. Leonard and H. Feder, "A computationally efficient method for large-scale concurrent mapping and localization," *Proc. IRSS,* p. 2, 1999.

[19] J. Guivant and E. Nebot, "Optimization of the simultanous localization and map building algorithm for real time implementation," *Transaction of Robotics and Automation,* p. 15, 2001.

[20] M. Montemerlo and S. Thrun , "Simultaneous localization and mapping with unknown data association using FastSLAM," *Springer Tracts in Advanced Robotics,* p. 28, 2001.

[21] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultanous Localization and Mapping that Provably Converges," *Proceedings of IJCAI,* p. 6, 2007.

[22] TurtleBot webpage, [Online]. Available: https://www.turtlebot.com/

Please note!
This Document is for information purposes only and any information given herein shall in no event be regarded as a warranty, guarantee or description of any functionality, conditions and/or quality of our products or any suitability for a particular purpose. With regard to the technical specifications of our products, we kindly ask you to refer to the relevant product data sheets provided by us. Our customers and their technical departments are required to evaluate the suitability of our products for the intended application.

We reserve the right to change this document and/or the information given herein at any time.

Additional information
For further information on technologies, our products, the application of our products, delivery terms and conditions and/or prices, please contact your nearest Infineon Technologies office (www.infineon.com).

Warnings
Due to technical requirements, our products may contain dangerous substances. For information on the types in question, please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by us in a written document signed by authorized representatives of Infineon Technologies, our products may not be used in any life-endangering applications, including but not limited to medical, nuclear, military, life-critical or any other applications where a failure of the product or any consequences of the use thereof can result in personal injury.