

Laser sensing from small UAVs

Michael Tulldahl^{*a}, Max Holmberg^b, Oskar Karlsson^b, Joakim Rydell^b, Erika Bilock^b,
Linnéa Axelsson^a, Gustav Tolt^b, Jan Svedin^c

FOI – Swedish Defence Research Agency, ^aDepartment of Electrooptical Systems, ^bDepartment of Sensor Informatics, ^cDepartment of Radar Systems, P.O. Box 1165, SE-581 11 Linköping, Sweden

ABSTRACT

Currently there is a considerable development of small, lightweight, lidar systems, for applications in autonomous cars. The development gives possibilities to equip small UAVs with this type of sensor. Adding an active sensor component, beside the more common passive UAV sensors, can give additional capabilities. This paper gives experimental examples of lidar data and discusses applications and capabilities for the platform and sensor concept including the combination with data from other sensors. The lidar can be used for accurate 3D measurements and has a potential for detection of partly occluded objects. Additionally, positioning of the UAV can be obtained by a combination of lidar data and data from other low-cost sensors (such as inertial measurement units). The capabilities are attainable both for indoor and outdoor short-range applications.

Keywords: Lidar, UAV, SLAM, Synthetic aperture radar

1. INTRODUCTION

The use of high-resolution laser scanners onboard small UAVs is becoming increasingly feasible. Many of these sensors, also referred to as lidar (light detection and ranging) sensors, are using the time-of-flight of several, simultaneously emitted, laser pulses in different directions to obtain accurate range measurements at high repetition rates. Range measurements (up to several hundred meters) together with compact scanning devices and electronics make these laser sensors an interesting complement to other more established sensors for small UAVs, such as visual and thermal cameras.

In this paper we present methods and experimental results for the integration of laser scanning in small UAVs. The purpose of the paper is to describe a sensor system operating in real-time, including hardware, software, and basic principles for the positioning of the sensor system. Experimental data is presented for two typical use cases; operation close to ground and operation using the lidar as a down-looking sensor at a higher altitude. Examples of collaborating sensors are given and a discussion is included on how the capabilities from the lidar sensor can complement other sensors.

The specific application of lidar data for positioning of the platform is the main focus of this paper. The ability to accurately determine the position of a platform, such as an unmanned ground or air vehicle (UGV or UAV), is often crucial in order to fully utilize sensor data from the platform. It is also necessary for enabling autonomous platform operation. Position information with sufficient accuracy can in many cases be obtained from receivers for GNSS (Global Navigation Satellite System) signals. There are, however, cases where the position accuracy provided by GNSS is insufficient. Additionally, in many environments, e.g., in forests, dense cities, indoors, and during adversarial jamming of GNSS signals, GNSS availability is reduced or absent. Therefore, complementary positioning systems are needed. Three examples of positioning techniques are:

- Inertial positioning: integration of sensed accelerations and rotations.
- Positioning based on signals of opportunity: detection and tracking of features not known to the navigation system in advance, e.g. radio sources, visual landmarks, geometric features etc.
- Infrastructure-based positioning: detection and tracking of beacons, reference markers, etc., whose positions are known in advance. GNSS is an example of infrastructure-based positioning.

^{*}michael.tulldahl@foi.se; phone +46 13 378516; foi.se

Most positioning systems combine more than one of these techniques; specifically, it is common to combine inertial navigation with either signals of opportunity or infrastructure.

Section 2 of this paper describes two typical use cases for lasers on small UAVs. The first case deals with the situation when the sensor platform operates close to ground or in indoor environments. An example of a system setup with hardware, software framework, and algorithms for positioning is given together with sample results. In the second case, results are shown and briefly discussed for operation at a higher altitude. In Section 3 we briefly describe three examples of potential collaborating sensor applications; positioning based on passive imaging, imaging with synthetic aperture radar technology, and a platform with multiple sensor systems. The current results, further development and combined use of several different sensors are discussed in Section 4. Concluding remarks are given in Section 5.

2. LASER SENSOR SYSTEM SETUP AND SENSOR POSITIONING

In this section we present two different methods regarding using laser systems onboard small, lightweight UAVs.

2.1 Operation close to ground or in indoor environments

Data for this part of the study were collected with a positioning system called Grape (GRAPh-based Precision lasEr), consisting of a 3D lidar, a flight controller, and software algorithms. The different versions of the firmware and software parts are shown in Table 1. The lidar was an OUSTER-OS1 with 16 channels, which collects approximately 330 000 points per second with a 33.2° vertical and 360° horizontal field of view. The lidar was tilted forward to get more points on the ground close to the lidar. The scanner rotation speed was set to 10 rotations per second. A Pixhawk Mini flight controller was used to obtain accelerometer and gyroscope data from its inertial measurement unit (IMU). The Raspberry Pi 4 Model B running Ubuntu Server was used for data collection and processing. The different components were mounted together with batteries on a wooden platform. This enabled carrying the system by hand, holding it above the head to simulate the movement of a UAV without having to fly.

The specific software frameworks used for the algorithm are Robot Operating System (ROS), Point Cloud Library (PCL), Eigen and Georgia Tech Smoothing and Mapping (GTSAM). ROS is a modular software infrastructure that eases the use of multiple sensors while enabling easy parallel development. The framework consists of nodes that send and receive data using topics. ROS enables using existing software from other developers, such as MAVROS¹ for communicating with the Pixhawk and the OS1 ROSnode² for easy communication and data transfer from hardware to software. ROS also have some other advantages, such as its built-in visualization tool, Rviz, which can be used to visualize trajectories and point clouds. Another useful feature is the built-in data saving tool, rosbag, which can be used to record and replay data streams. Point Cloud Library is a useful framework for managing and processing large data structures such as 3D lidar data.³ In this study PCL was used to store and transfer lidar point clouds between ROSnodes by converting between PCL point clouds and ROS point clouds. PCL was also used for transformation of point clouds between different coordinate systems, and for k-d tree searching methods. Eigen is a C++ library which was used for linear algebra operations with varying matrix sizes.⁴ GTSAM is a C++ library used for robotics and computer vision applications by implementing sensor fusion based on factor graphs and Bayes networks.⁵

Table 1. Versions of the different firmware and software used for the GRAPE system.

Name	Ubuntu Server	Ouster firmware	Pixhawk mini firmware	ROS	GTSAM	PCL	Eigen
Version	18.04	1.13.0	Arducopter 4.0.2	Melodic Morenia	4.0.2	1.8.1	3.3.4

The type of positioning algorithm used is called Simultaneous Localization And Mapping (SLAM). SLAM algorithms use the principle of creating some type of map and simultaneously localizing a platform/sensor/robot within that map. This particular SLAM algorithm is a combination of inertial positioning and positioning based on signals of opportunity from the lidar.

Graph-based SLAM has been a popular method of formulating the SLAM problem for some time. One of the first publications to formulate SLAM as a set of links between robot poses was Ref. [6] and many have since followed e.g. Refs. [7,8,9,10]. Graph-based SLAM has a number of advantages over the more traditional filtering techniques, key among them is the fact that older measurements are not discarded and can be taken into consideration at each time step. Today graph-based SLAM can be considered one of the standard ways of formulating the SLAM problem. Here follows a very brief explanation of the basic concept of graph-based SLAM, further reading can be found in Refs. [11,12,13]. Consider Figure 1, and let X_k represent the robot position at time k and let L_n represent a landmark with index n . The odometry and landmark measurements O_{12} , O_{23} , and Z_{21} , Z_{22} , Z_{32} are called factors and relate the states to each other with a certain probability. We are interested in the states X_1, X_2, X_3 given the factors O_{12} , O_{23} , and Z_{21} , Z_{22} , Z_{32} , which can be expressed as the following probability,

$$P(X_1, X_2, X_3 | O_{12}, O_{23}, Z_{21}, Z_{22}, Z_{32}) \propto f(X_1, X_2, X_3), \quad (1)$$

where $f(X_1, X_2, X_3)$ is the value of the factor graph. In order to find the most likely trajectory of the sensor, a maximum a-posteriori inference is done by maximizing this probability. Due to the inherent nonlinearities in a moving robot, this maximization problem boils down to a nonlinear optimization problem that can be solved with a nonlinear optimizer. The corresponding information matrix to a factor graph generated by a robot moving through an environment is in general sparse. This sparsity can be exploited when solving for the optimal trajectory and is the reason why modern graph-based SLAM algorithms can work in real-time.

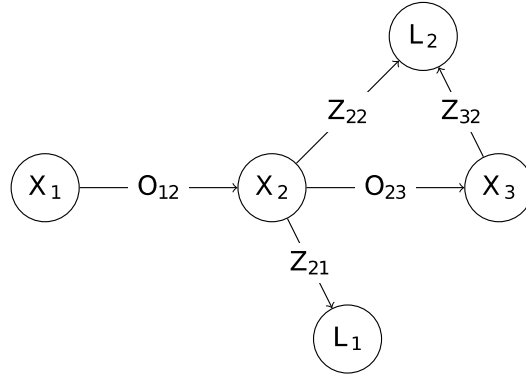


Figure 1. A small factor graph with robot poses represented as X_i , and landmark poses as L_n connected by odometry factors O_i and landmark measurement factors Z_i .

The algorithm, developed for close to ground and indoor positioning, is divided into four different components; feature extraction, feature association, sensor fusion, and a precision node. These components and how they interact are displayed in Figure 2. In the following paragraphs, each of these components are explained in more detail.

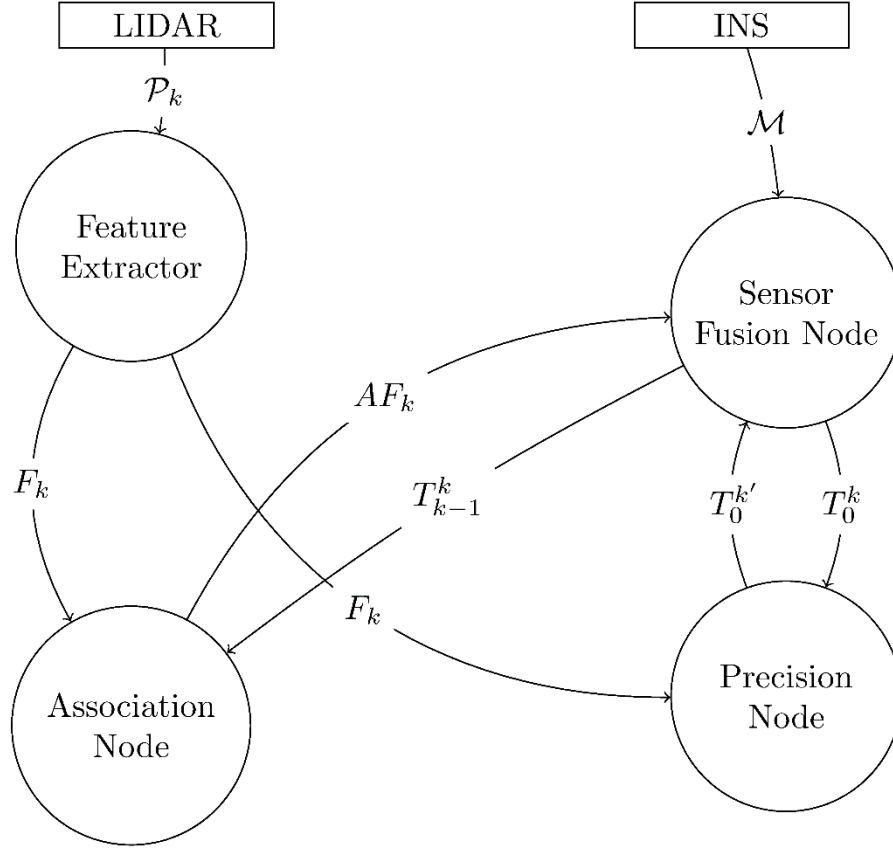


Figure 2. A graphical representation of the algorithm, where P_k is the raw point cloud at time k from the lidar, F_k are the extracted features, AF_k are associated features, M are accelerometer and gyro measurements, T_{k-1}^k is the transform between the robot pose at time $k-1$ and the robot pose at time k , T_0^k is the transform between the robot pose at time 0 and the robot pose at time k , $T_0^{k'}$ is the updated transform between the robot pose at time 0 and the robot pose at time k .

The *feature extraction* is performed by first filtering out candidate features, planes and edges, from each lidar scan.¹⁴ The filter operation is fast, because it is applied on the structured point cloud, ordered along the lidar azimuth and elevation directions, and thus does not require neighbour search operations. The feature candidates are then examined in a local region-of-interest (ROI) around each feature candidate. The ROI-points for each plane feature candidate are fitted to a plane and the candidate with lowest point-to-plane root-mean-square error (RMSE) distance is selected. Similarly, edge candidates are fitted to lines, and the candidate with lowest point-to-line distance RMSE is selected. The scan frame is divided into four sectors (right, back, left, forward) to select one plane and one edge in each sector.

Once the features, F_k , have been extracted from a lidar-scan they need to be *associated* to the previously constructed map. The feature point cloud is first transformed to the currently best-known position. Thereafter a k-d tree is used to find the k nearest neighbours to a feature measurement based purely on the Euclidean distance between feature centre points. Then, if the normal of a new feature matches the normal of a feature in the map, the feature is associated, and receives a global feature index. The associated features, AF_k , are then passed along to the sensor fusion part of the algorithm.

The *Sensor Fusion Node* uses the preintegrated IMU Factor included in the GTSAM library.^{15,16,17} The planar and edge features are represented in the graph using infinite planes and lines in 3D. The backend of this graph-based SLAM algorithm is created with the GTSAM library. Specifically the algorithm called ISAM2¹⁸ is used in order to create and optimize the factor graph in real-time. ISAM2 is an incremental SAM (Smoothing And Mapping) algorithm that utilizes the Bayes tree for efficient mapping.¹⁹ Since the SLAM algorithm needs to run in real-time on a lightweight platform, a Raspberry Pi 4, it is essential that the sensor fusion computation time is kept to a minimum. As the graph grows in size as

more and more features are added, the allowed computation time for a scan will eventually be exceeded, making pruning of the graph essential. In order to keep the graph at a manageable size the oldest features in the graph are removed as needed.

The *Precision Node* is used to counteract drift primarily in the x-, y-, z-directions, and around the yaw axis. The roll and pitch drift is continuously held at low values based on information of the direction of the gravitational force (from the IMU), which is fed into the Sensor Fusion Node. The Precision Node combines features from several scans (we use five scans) and optimizes the corresponding poses with a rigid transform. The precision-part of the algorithm has been used in earlier work e.g. in Refs. [14,20] and has two important properties, which improves the accuracy. First, it reduces the pose error, caused by inaccurate individual range measurements, because more features are available from several combined scans. Secondly, the node weighs associated features by age, applying higher weight to old features, which reduces accumulation of errors over time. For the precision node, all earlier features are made available for association. The Precision Node output, with the most accurate pose estimate, is fed into the Sensor Fusion Node.

In order to get rudimentary quantitative measurements of the SLAM algorithm accuracy, we created datasets where the actual start and end positions and orientations were the same, to an accuracy of about 5 cm and 2°. We then used the error between the SLAM-algorithm start and end poses divided by the estimated total distance as our accuracy measurement. The orientation error was only measured in the yaw direction of the platform, and was calculated by dividing the final error in yaw angle compared to the start orientation with the sum of the total yaw-rotation of the platform. The total yaw-rotation was counted in ticks of 45 degrees, rotations smaller than 45 degrees did not contribute to the total sum. This was done in order to reduce the small rotations generated by walking with the platform. The results are summarized in Table 1 for four indoor and two outdoor environments. Multiple floors were traversed in indoor environments in datasets 1 and 6. Examples of estimated sensor trajectories and lidar point clouds are shown in Figure 3.

As shown in Table 2, the resulting position and yaw errors varies between the different types of environments. Datasets 1 and 6 have very low errors whilst the errors in datasets 2 and 3 are relatively high. This high variation between datasets indicates that the algorithm has a very low drift as long as it does not “get lost” i.e. it cannot find features in its current environment or cannot lock on to the previous features for an extended period of time. Dataset number 2 consisted of long narrow corridors, which proved to be a difficult problem due to a lack of features that could be observed for a long period of time and the fact that corridors are very symmetrical and similar for long stretches, increasing the risk of false associations. In general, the outside environments are more difficult due to a lower feature density. For feature-rich indoor environments the position accuracy of our method agrees and performs well compared to the LOAM (Lidar Odometry and Mapping) method published by Zhang and Singh,¹⁴ where position errors of 0.9% and 1.3% were reported for two different indoor environments. Some parts of our algorithm are similar to the LOAM method, such as the plane and edge extraction and the precision node (called *mapping* in LOAM). However, as the lidar sensor, parts of the algorithm, and environments differs in the LOAM study, accuracy numbers are not directly comparable.

Table 2. Quantitative measurements of the SLAM algorithm in both outdoor and indoor environments.

Dataset	Environment	Total travelled distance	Position error		Total twisted yaw	Yaw error	
		(m)	(m)	(%)	(°)	(°)	(%)
1	Indoor stair case	110	0.55	0.5	2430	2.3	0.094
2	Indoor corridors	204	13.2	6.5	540	9.2	1.70
3	Parking lot with open spaces	108	6.2	5.7	450	2.3	0.51
4	Cluttered outside environment	221	5.1	2.3	540	2.9	0.53
5	Indoor one floor	79	1.4	1.8	810	1.4	0.17
6	Indoor two floors	97	0.2	0.2	1080	0.5	0.05

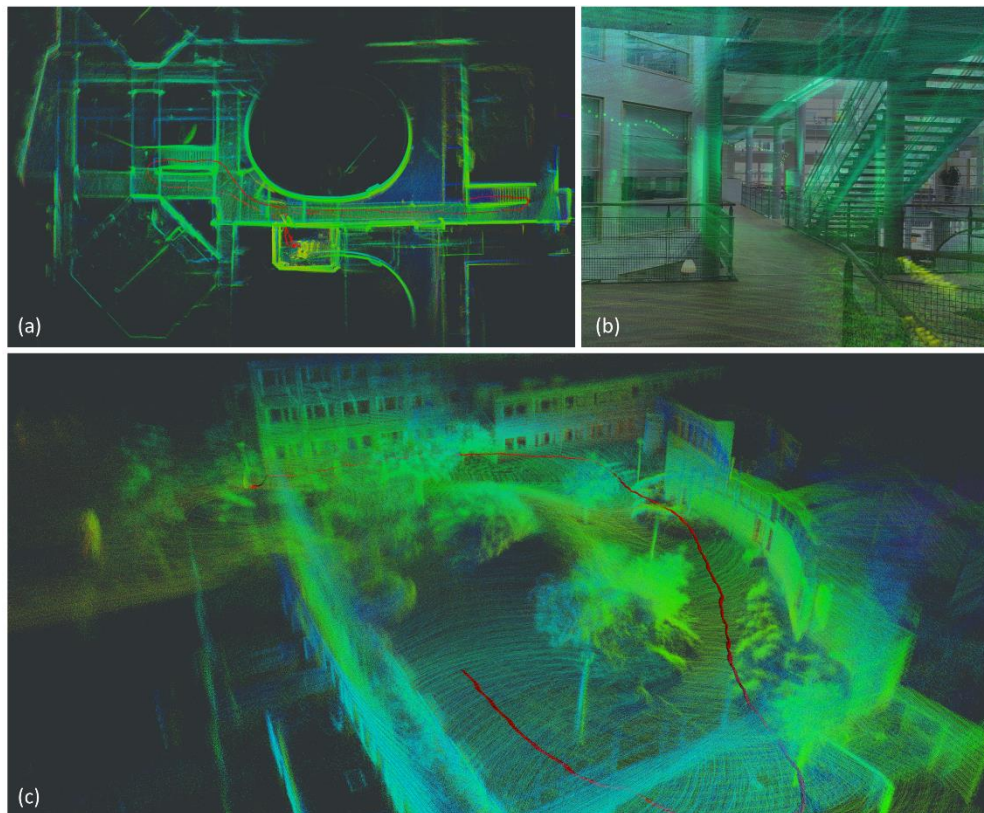


Figure 3. Examples of lidar point clouds from indoor (a-b) and outdoor (c) environments collected from an altitude of about 2 m. Figure (b) shows an overlay detail of the lidar point cloud on a photograph of the indoor environment. The sensor trajectory is indicated in red (a) and (c), or with green large dots (b). Data was collected from a development platform carried by one person.

2.2 Operation as a down-looking sensor

Figure 4 shows one of our UAV test platforms equipped with a Velodyne VLP-16 lidar sensor. Similarly to the OUSTER-OS1, the VLP-16 is a rotating scanning lidar with 16 channels which collects approximately 330 000 points per second. In this setup, the sensor is mounted in a down-looking direction, so that the rotating scanner sweeps the scan lines on the ground in a pattern perpendicular to the forward flight direction. The sensor is tilted slightly in the forward direction to obtain an almost vertical direction of the center lidar ray when the multirotor UAV is pitching forward in forward flight.

Operation of the lidar as a down-looking sensor changes the situation both for positioning and for other applications, compared to horizontal operation. The positioning of the sensor based on lidar data generally becomes more difficult in the horizontal plane (x- and y-planes) because lidar range measurements to landmarks are not performed parallel to the horizontal plane. The ability to detect objects might however become easier and more efficient, because of the larger viewing area exposed to the sensor. For a rotating lidar sensor, like the Velodyne VLP-16 or OUSTER-OS1, the downward mounting angle has the consequence that emitted laser pulses, with scanner angles directed to the sides or upwards, do not give useful range measurement data. We have found that scanner angles at least up to 45° , on each side from the vertical, give useful data. Consequently, the Velodyne VLP-16 collects at least 80 000 points per second from ground or from objects or vegetation on ground in this case.

In Figure 5 we show examples of lidar point clouds, collected from an altitude of about 15 m above ground over a birch tree area. The SLAM method used to assemble the point cloud was not the same as the method described in subsection 2.1. Instead of using landmarks (features), we used all points from each scanner frame and registered those against a ground model, based on data from earlier frames. The method is described in more detail in previous publications.^{21,22} Although the resulting accuracy is relatively high, the method has the drawback of being computationally expensive and is not implemented for onboard real-time operation on the platform.

There is a high potential to detect and possibly classify objects with the down-looking lidar sensor. The primary source of information is the 3D data, but also intensity data (estimated reflectance) is of interest for detection algorithms. The sensor is able to work both during day- and night-time and has the ability to penetrate through obscurants. In Figure 5 (b) the ground points are color-coded with the information from the reflected lidar pulse intensity, from brown for low intensity to white for high intensity. Although some noise in the intensity data is present, there is e.g. a clear difference between pulses reflected from the grass, the car, and the small trench shown in the photograph in Figure 5 (d). The low-intensity pulses reflected from the car are caused both by its dark color and by the specular component of the paint. The penetration ability of the laser pulses through vegetation can be observed in Figure 5 (b). It is to be noted that the point density on the ground and the reflected pulse intensity is lower close to, and in the shadow of the trees. The lower intensity in the shadowed areas is caused by the birch trees (leaves or other parts) which subtend parts of the laser beam footprint while other parts of the same beam continues to the ground, which reduces the ground pulse intensity.

Figure 6 shows another example of a lidar point cloud where the color scale represents altitude in Figure 6 (a) and lidar intensity in Figure 6 (b). From the control tower, shown in Figure 6 (c), runs an overhead line indicated with an ellipse. The corresponding point data in Figure 6 (a-b) illustrates the important capability of the lidar sensor to detect and localize narrow objects. This capability can be difficult to achieve for passive visual sensors, especially if the narrow object is imaged against a background with similar reflectance as the object.



Figure 4. Photograph of the UAV equipped with a Velodyne VLP-16 lidar.

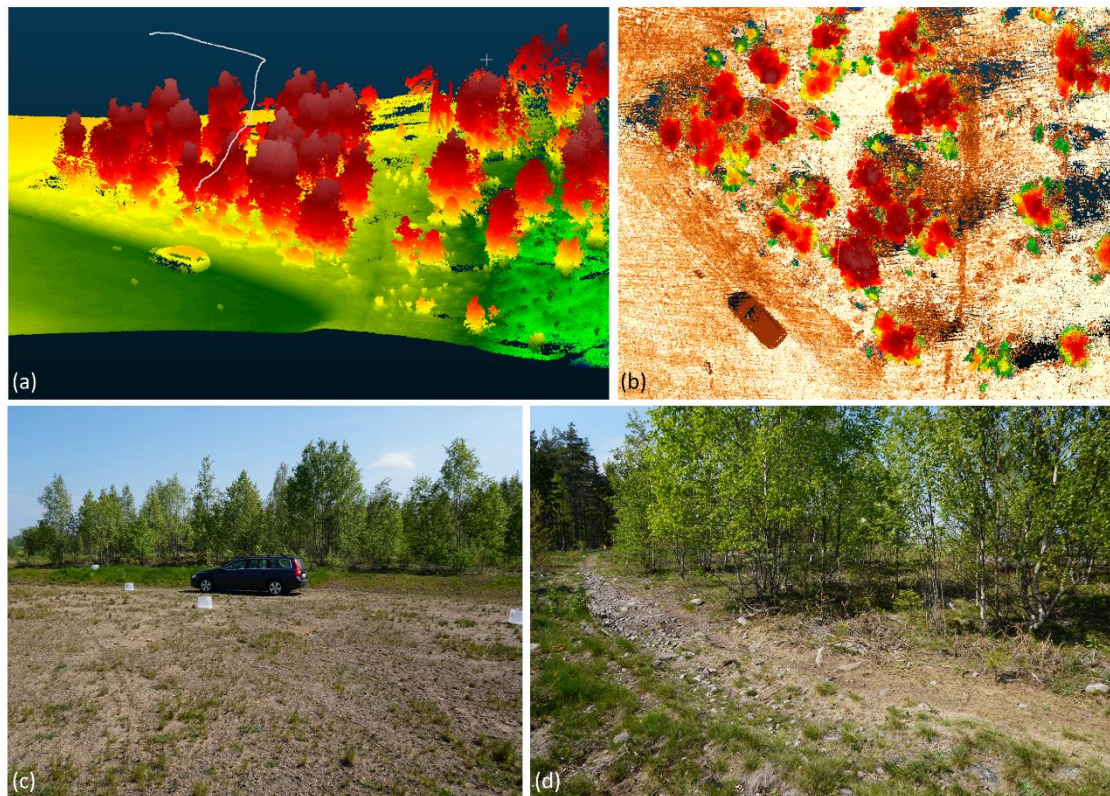


Figure 5. Examples of lidar point clouds collected from an altitude of about 15 m (a-b) above ground over a birch tree area. In figure (a) the points are color coded based on altitude and the UAV trajectory in white. Figure (b) shows lidar points from trees and high vegetation color coded based on altitude, and ground points coded based on intensity, from brown (for low intensity) to white (for high intensity). Figures (c) and (d) show photographs of a car in front of the tree area and a small trench on the backside of the tree area. The lidar data was collected from the UAV shown in Figure 4.

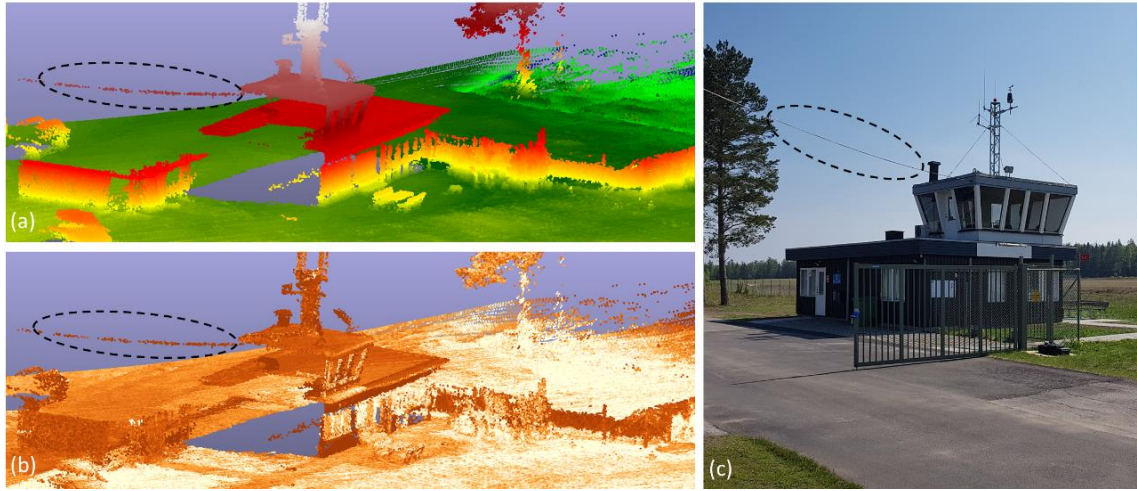


Figure 6. Examples of lidar point clouds collected from an altitude of about 15 m above ground over an airfield control tower (a-b). In figures (a) and (b) the points are color coded, respectively, with altitude and intensity (brown for low intensity, white for high intensity). Figure (c) shows a photograph of a part of the scene with an overhead line indicated with an ellipse. The line is also marked in point data in figures (a) and (b). The lidar data was collected from the UAV shown in Figure 4.

3. POTENTIAL COLLABORATING SENSORS

There is a number of potential sensors that could be combined with laser sensors for improved capabilities. In this section we briefly describe three examples from our ongoing work. The three examples are positioning based on passive imaging, imaging using synthetic aperture radar technology, and a platform with multiple sensor systems.

3.1 Positioning based on passive imaging

The Kiwi, shown in Figure 7 (a), is a real-time system for indoor positioning and mapping, which utilizes inertial navigation combined with tracking of visual features. It is the latest member of the family of *Chameleon* positioning systems,^{23,24,25} but is built using small and more light-weight components, in order to enable mounting on a small UAV for indoor operation. The Kiwi navigates by detecting recognizable points (often referred to as *landmarks*) in images from a stereo camera, i.e. two cameras oriented in the same direction, separated by a small distance from each other. The Kiwi tracks these points over time as the system moves through an environment. The inertial sensors, which measure acceleration and rotation, help predict where tracked points should appear in a new image, thereby improving the tracking accuracy. The position and orientation of the camera are estimated based on the tracked landmarks.

Since a stereo camera is used, the approximate distance to most points in the observed scene can be estimated from a single image pair. By combining these measurements with the camera position and orientation along a trajectory, a 3D point cloud representation of the environment is generated. Figure 7 (b)-(c) shows point clouds, generated while moving through an indoor environment.

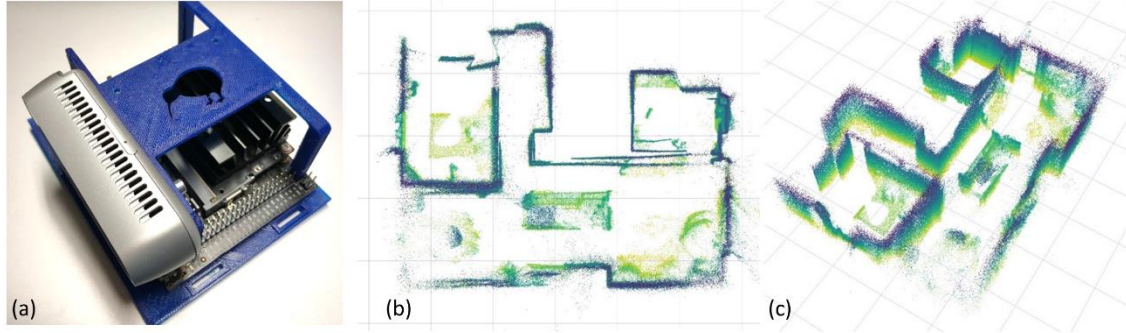


Figure 7. The Kiwi, a real-time indoor positioning system, comprised of a NVIDIA Jetson Nano computer and an Intel RealSense D435i stereo camera with inertial sensors (a). Figures (b) and (c) show top view and oblique view of a point cloud model estimated by the Kiwi. Colors indicate altitude (blue: high, yellow: low).

We have developed another system called the Snowy owl, a down-looking image-based positioning system for UAVs, which can manage without GNSS. The UAV used in our field trials can be seen in Figure 8 (a). It is a quadcopter, with the Snowy Owl system, consisting of an onboard computer and a gimbal with a camera. Briefly, the system uses the onboard camera to take images, which are compared to a reference image (i.e. a satellite image over the operational area). The UAV position is estimated as one of the positions, in the reference image, which matches the camera image. Since there may be several matches, the actual position of the UAV is determined by a particle filter, which fuses the image matching result with the UAV velocity. The velocity is estimated using optical flow; the movement for each pixel between consecutive images. See Ref. [26] for a more detailed description of the algorithm.

The Snowy Owl algorithms run in real-time, like the Kiwi system and the Grape system described in subsection 2.1. We use the estimated position to control the quadcopter during waypoint missions and other autonomous flight. The positioning method is robust to small differences between the camera image and the reference image, i.e. lighting conditions and some seasonal variations. Significant changes like snowfall and construction work would demand a new or updated reference image.

Our GNSS-independent positioning algorithm is integrated with a method for detecting and following moving objects during flight. The algorithm works by detecting and tracking points in the image where both the optical flow and intensity deviate from their surroundings. Since the optical flow is calculated between consecutive images, objects moving relative to the surrounding area will cause optical flow deviations. However, deviations are also caused by objects at different altitudes, and some objects that are higher than their surroundings also meet the requirement of a deviant intensity. These objects are sorted out using an object tracking algorithm, where only tracks which move a certain distance between images are accepted. Image samples showing the positioning and target tracking calculated by the Snowy Owl are shown in Figure 8 (b)-(c) together with images taken by the onboard camera at the same times in Figure 8 (d)-(e).

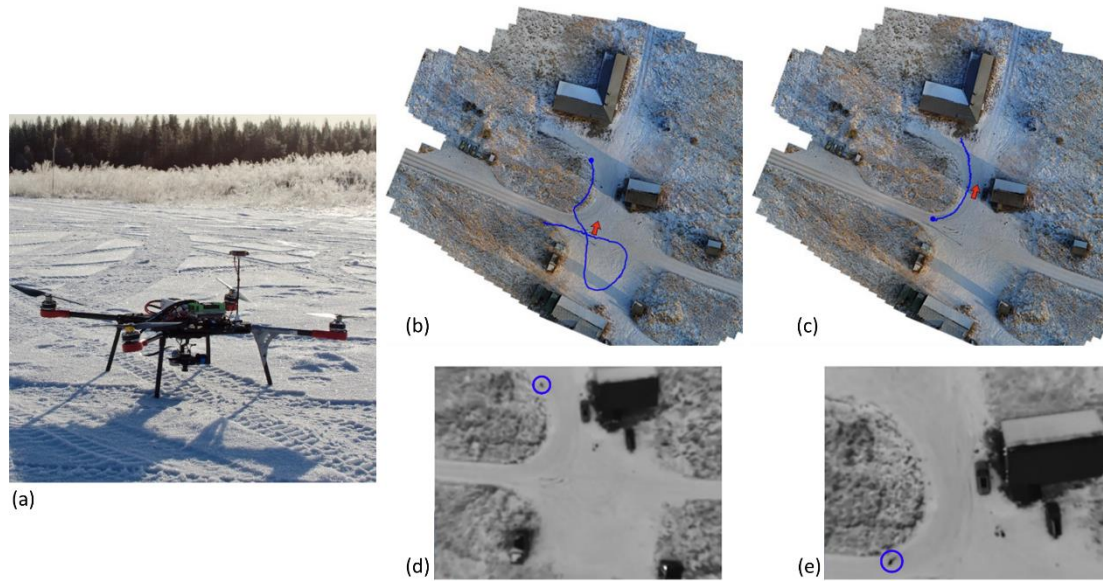


Figure 8. The Snowy Owl positioning system mounted on a quadcopter (a). Figures (b) and (c) show reference images with red arrows indicating position estimates from the Snowy Owl overlaid. Blue lines in the images are estimated target tracks calculated by the Snowy Owl. Figures (d) and (e) show images from the camera onboard the Snowy Owl. Blue circles are manually marked targets, to illustrate input to the tracking algorithm. Left and right image pairs show examples at different times.

3.2 Synthetic aperture radar

Synthetic Aperture Radar (SAR) has many applications in fields such as terrain mapping, environmental monitoring, astronomy, law-enforcement, and military surveillance. A SAR sensor can achieve high resolution at long range without a large physical aperture, and is less sensitive to weather conditions and time of day compared to an electro-optical sensor.

A SAR image is formed by combining radar measurements from different positions along a flight trajectory. This effectively synthesizes a virtual antenna that is much larger than the physical antenna used. Accurate focusing requires good knowledge of the relative positions of the measurement points. In theory such knowledge could be acquired by using a perfectly controlled pre-defined flight path. However, as a small UAV may travel along an irregular path involuntarily due to unexpected winds, suboptimal control, etc., position measurements are required for well-focused SAR images. Accurate position measurements can e.g. be achieved using a GNSS/RTK (Real-Time Kinematic) sensor, where a ground based GNSS sends corrections to a GNSS system onboard to achieve an accuracy much better than for a stand-alone GNSS sensor. Recently, several low-cost GNSS/RTK systems have become commercially available. As the UAV GNSS and radar antenna phase centres typically do not coincide, an attitude sensor can additionally be used to more accurately determine the relative position offset of the GNSS and radar antenna phase centers. This is mainly important for non-linear flight paths.

We have successfully tested a strip-map SAR system assembled on a small UAV using a low-cost 5-6 GHz radar, GNSS/RTK, and IMU sensors. A photograph of the UAV used, and a schematic of the system architecture is shown in Figure 9. An example of a generated SAR image is shown in Figure 10, which also shows a photo of the scene from a visual camera on the UAV. In this experiment a zigzag flight path was used. The achieved image resolution (for a corner reflector) was found close to the theoretical limit in both on-track and cross-track directions.

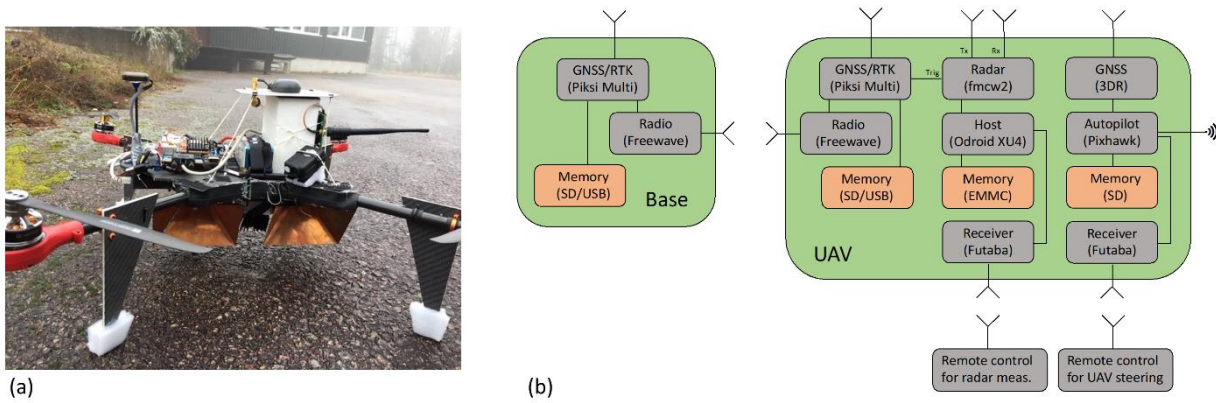


Figure 9. Photograph of the UAV equipped with SAR measurement system payload (a) and a schematic illustration of the system architecture (b).

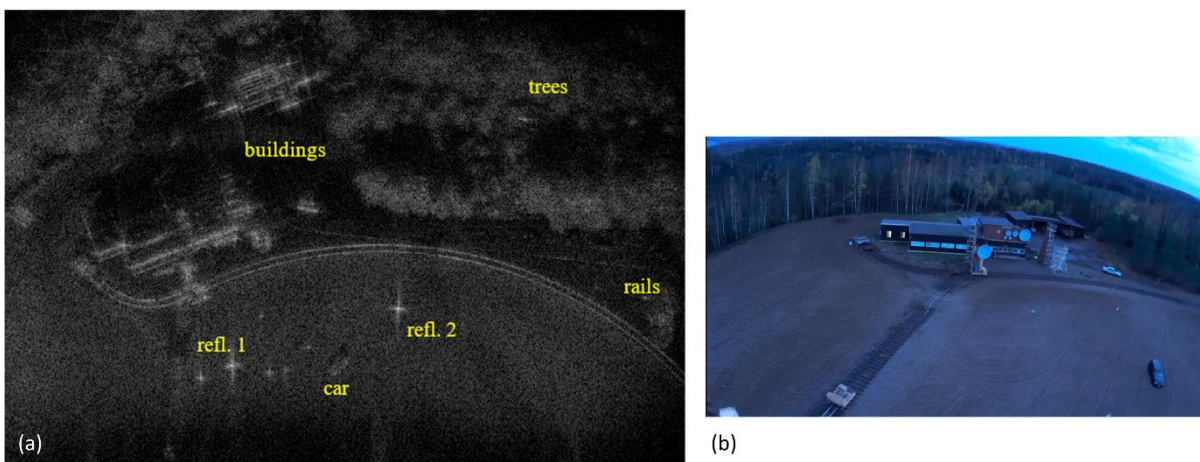


Figure 10. Generated SAR image (a) of the scene shown in the photograph (b). The measurement altitude was approximately 28 m and the length of the flight path about 150 m. A zigzag flight path was used to test the accuracy of the position and attitude sensors. The photograph of the scene was taken from a visual camera mounted on the UAV.

3.3 Multi-sensor UAV platform

The challenge with unreliable communication links together with low or no GNSS-signal, and navigation through narrow environments is a situation we are currently exploiting with the MAX-drone shown in Figure 11. The Multi-purpose Autonomous eXploring drone (MAX) is being developed in the Horizon 2020 project INGENIOUS. MAX is intended as a first responder's extra companion that assists in the exploration and assessment of potentially dangerous buildings by sending multi-sensor data to a ground control station for manual inspection and automatic analysis. Using a variety of complementary sensors and an onboard distributed computational system, MAX will build and update a 3D representation of the environment, determine waypoints for optimal exploration, find openings large enough to pass through, plan routes and avoid obstacles, in order to navigate safely through complex and narrow geometry and under varying lighting conditions. The intended use requires the UAV to fly indoors or close to ground. In this context a possible solution is to combine both the lidar-based positioning and visual stereo camera positioning described in subsections 2.1 and 3.1. The lidar-based positioning has in this case the advantage of a larger maximum range to landmarks than the stereo camera. Using both systems has the potential of achieving a more robust positioning, which is important for safe navigation in narrow and complex geometries. With several sensors supporting the navigation, also the capability to avoid obstacles is improved. In addition to electro-optical sensors, sonars can be used to detect small objects such as wires and ropes as well as glass windows, and assist with close-range collision avoidance.

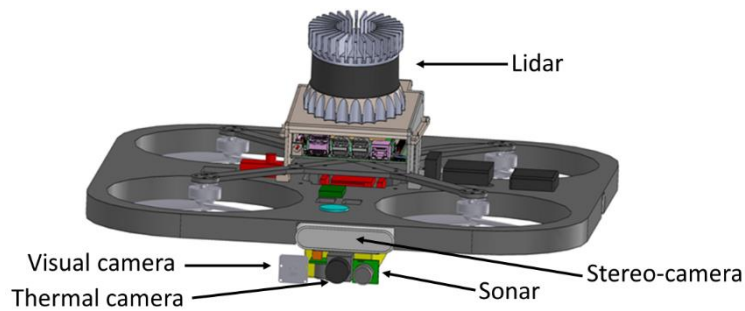


Figure 11. CAD-image of the Multi-purpose Autonomous eXploring drone.

4. DISCUSSION

The combination of multiple sensors in one package gives the advantage of being able to handle multiple environments and scenarios. Beside thermal cameras, not treated in this paper, lidar and SAR have the advantages of working in darkness. Lidar and camera-based positioning systems could be combined for a potentially more robust and accurate positioning solution than would be possible using separate systems. Lidar and SAR can be used for mapping and detection of occluded objects which can be difficult for visual cameras. Two downsides of having multiple sensors are the added weight to the platform, with reduced flight time, as well as limiting the processing power onboard for the sensor systems to be used.

The ability to run real-time positioning systems and data processing on smaller processing units such as Raspberry Pi and Jetson Nano enables the development of self-sufficient UAVs. Running the systems onboard gives the potential of being independent of GNSS data or constant communication with a ground unit.

There are limitations to communications when using multiple sensors with high throughput such as lidar and visual cameras. This limitation can be handled by processing data onboard and only communicating the most necessary information such as control commands to the UAV and e.g. detected objects to ground. There are also situations when the communication link between the ground station and the UAV is broken or temporarily interrupted. These problems can occur when flying outdoors with no cell coverage or indoors behind multiple walls, where neither Wi-Fi nor cell phone signals can reach. To handle this, a combination of dedicated links, cell phone networks, and Wi-Fi networks could be used, as well as temporarily buffering data before communicating to the operator.

5. CONCLUSIONS

We have presented laser systems, including hardware and software algorithms, for positioning of small UAVs. The methods presented for operation indoors or close to ground are capable of operating in real-time with a small onboard computer. This is an important capability, especially to enable autonomous operation of platforms, which relies on accurate positioning.

The laser sensor has the advantage of producing accurate range and angular measurements to achieve both accurate positioning of the platform and potentially mapping and detecting objects in dark, shadowed areas or through obscuring such as vegetation.

Future multi-sensor platforms can include both electro-optical and other sensors, such as sonars and radars. The most efficient combination of sensors depends on the operational context, the altitude, as well as size, weight and power requirements. The use of high-resolution active sensors, such as radar sensors and scanning laser systems, are presently rare on small UAVs. With continued development of sensors, electronics, integration and algorithms, more capable small UAVs are expected in the future, both in terms of autonomous behavior and in survey and object detection capabilities.

ACKNOWLEDGMENTS

The work presented in this paper has been supported by the Swedish Armed Forces, the Swedish Defence Materiel Administration (FMV) on behalf of the Swedish Armed Forces, has received funding from Visual Sweden and the Swedish Innovation Agency (VINNOVA), and has received funding from the European Union's Horizon 2020 Research and Innovation Programme and the Korean Government under Grant Agreement No 833435. Content reflects only the authors' view and European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

-
- [1] Vladimir Ermakov, "Package Summary", ROS, 3 Mars 2018, <http://wiki.ros.org/mavros> (13 August 2020).
 - [2] "OS1 Example Code", github, 17 Mars 2020, https://github.com/ouster-lidar/ouster_example (13 August 2020)
 - [3] "About," Point Cloud Library, <https://pointclouds.org/about/> (13 August 2020).
 - [4] "Main page", Eigen, 29 July 2020, http://eigen.tuxfamily.org/index.php?title=Main_Page (13 August 2020).
 - [5] "GTSAM 4.0 Factor graphs for Sensor Fusion in Robotics", GTSAM, <https://gtsam.org/> (13 August 2020).
 - [6] Lu, F., and Milios, E., "Globally consistent range scan alignment for environment mapping," *Autonomous robots* 4, 333-349 (1997).
 - [7] Dellaert, F., and Kaess, M., "Square Root SAM: Simultaneous localization and mapping via square root information smoothing," *The International Journal of Robotics Research* 25, 1181-1203 (2006).
 - [8] Thrun, S., and Montemerlo, M., "The graph SLAM algorithm with applications to large-scale mapping of urban structures," *The International Journal of Robotics Research* 25, 403-429, (2006).
 - [9] Grisetti, G., Stachniss, C., Grzonka, S., and Burgard, W., "A tree parameterization for efficiently computing maximum likelihood maps using gradient descent," *Proc. Robotics: Science and Systems* 3, (2007).
 - [10] Shan, T., and Englot, B., "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," *Proc. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4758-4765 (2018).
 - [11] Kschischang, F. R., Frey, B. J., and Loeliger, H. A., "Factor graphs and the sum-product algorithm," *IEEE Transactions on information theory* 47, 498-519 (2001).
 - [12] Loeliger, H. A., "An introduction to factor graphs," *IEEE Signal Processing Magazine* 21, 28-41 (2004).
 - [13] Dellaert, F., "Factor graphs and GTSAM: A hands-on introduction," Technical Report GT-RIM-CP&R-2012-002, Georgia Institute of Technology (2012).
 - [14] Zhang, J., and Singh, S., "LOAM: Lidar Odometry and Mapping in Real-time," *Proc. Robotics: Science and Systems Conference (RSS)*, (2014).
 - [15] Forster, C., L. Carlone, F. Dellaert and D. Scaramuzza. "IMU Preintegration on Manifold for Efficient Visual-Inertial Maximum-a-Posteriori Estimation." *Proc. Robotics: Science and Systems*, (2015).
 - [16] Carlone, L., Kira, Z., Beall, C., Indelman, V., and Dellaert, F., "Eliminating conditionally independent sets in factor graphs: A unifying perspective based on smart factors," *Proc. 2014 IEEE International Conference on Robotics and Automation (ICRA)*, 4290-4297 (2014).
 - [17] Lupton, T., and Sukkarieh, S., "Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions," *IEEE Transactions on Robotics* 28, 61-76 (2011).
 - [18] Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J. J., and Dellaert, F., "iSAM2: Incremental smoothing and mapping using the Bayes tree," *The International Journal of Robotics Research* 31, 216-235 (2012).
 - [19] Kaess, M., Ila, V., Roberts, R., and Dellaert, F., [The Bayes tree: An algorithmic foundation for probabilistic robot mapping], Springer, Berlin, Heidelberg, 157-173 (2010).
 - [20] Tulldahl, M., Rydell, J., Holmgren, J., Nordlöf, J., and Willén, E., "Lidar-based positioning in forest environments," *Proc. SPIE* 11160, 1116005-1-16 (2019).
 - [21] Tulldahl, H. M., Bissmarck, F., Larsson, H., Grönwall, C., and Tolt, G., "Accuracy evaluation of 3D lidar data from small UAV," *Proc. SPIE* 9649, (2015).
 - [22] Tulldahl, M., Larsson, H., Tolt, G., Bissmarck, F., Grönwall, C., and Nordlöf, J., "Application and capabilities of lidar from small UAV," *Proc. SPIE* 9832, 98320V-98320V-15, (2016).
 - [23] J. Rydell and E. Bilock, "Panther Chameleon: Real-time Image-based Positioning and Mapping," *Proc. IEEE/ION ITM*, (2015).

-
- [24] E. Emilsson and J. Rydell, "CHAMELEON on Fire - Thermal Infrared Indoor Positioning," Proc. IEEE/ION PLANS, (2014).
 - [25] J. Rydell, E. Bilock, and H. Larsson, "Navigate into Danger," Proc. IEEE/ION PLANS, (2016).
 - [26] J. Rydell, E. Bilock, M. Tulldahl, "Computationally Efficient Vision-based UAV Positioning," Proc. Institute of Navigation International Technical Meeting (ION ITM), (2019).