Indoor drone piloting:
A literature overview

Sebastian Wehkamp

Abstract—In recent years an increasing amount of research done has been done on the development of autonomous flying agent, however this research mostly focuses on flying outdoors where the systems can rely on absolute navigation systems like GPS. These navigation systems are not available indoors so the agents have to rely on other sensors to explore and navigate unknown indoor environments. Positioning methods can be split into two categories, absolute positioning methods and relative positioning methods. Absolute positioning methods use external devices to localise the agent while relative positioning methods only use devices located on the agent. A literature overview is presented of the available absolute and relative positioning methods together with their respective sensors. A simulation in which a flying agent performs localisation and navigation is executed in order to compare the advantages and disadvantages of various positioning methods focusing on the computational resources used. The simulation provides a modular framework where the simulation part can almost be completely replaced by the actual MAV. This could stimulate research in an area where not much research has been done, localising flying agents using absolute positioning methods or a combination of absolute and relative positioning methods. We expect that this can improve the performance of localisation techniques and that it will be an important future research topic.

Index Terms—Autonomous navigation, indoor localisation, relative positioning, absolute positioning.

1 Introduction

The increasing popularity of micro aerial vehicles (MAVs) resulted in usage in many different areas like surveillance [1], search and rescue missions [2], and photography [3]. In most cases these MAVs are controlled by human operators. In recent years the research community started showing interest in developing autonomous MAVs. Whereas most of the research is done on flying in outdoor environments, vehicles that can autonomously fly indoors are envisioned to be useful in a variety of applications including autonomous warehouse inventory, discovering faulty or missing parts in assembly lines, or for indoor transportation. The main advantage of MAVs over ground vehicles, which currently are often used, is their increased mobility [4].

As for ground vehicles, the main goal of an MAV is to reach the target location without human interference. In the literature, this is known as *navigation*. Navigating flying agents outdoors on obstacle-free heights can be done using a global navigation satellite system (GNSS) like the global positioning system (GPS). In indoor environments GNSS is often not available and not precise enough to allow for navigation through complex indoor 3D environments. To fly autonomously in indoor environments a variety of sensors are used. Depending on the methodology different sensors are needed to map the environment, perceive obstacles, avoid the obstacles, and plan a flight path [4]. Although the general principles of navigation can be transferred from ground vehicles to flying vehicles there are three reasons why this is not as straightforward. First there is the limited payload of an MAV, MAVs cannot carry the same variety of sensors as a ground robot. Second, MAVs have additional degrees of freedom which prevents the direct usage of efficient 2D algorithms for navigation. A third reason is that the dynamics of a flying robot are more complex than a ground robot which makes them harder to control [5].

Navigation systems can be classified into two categories, absolute positioning systems and relative positioning systems. [6]. Absolute positioning makes use of external devices to perform localisation of the MAV. These external devices are for example beacons or landmarks which are located throughout the environment. An example of an absolute positioning systems would be GPS which uses satellites as external devices. Other examples make use of ultrasound [6, 7, 8], Bluetooth [9, 10], or Wi-Fi [11, 12]. In contrast to these absolute positioning systems, relative positioning systems do not have to be installed in the area where the positioning system is served. These relative positioning systems make use of sensors placed on the MAV self [13]. An overview of the relative and absolute positioning systems discussed in this paper can be found in Figure 1.

This paper will contain a literature overview of research that has been done in the field of indoor drone navigation. This is done by describing the different approaches used throughout the research grouped into categories based on which sensors they use. The remainder of this paper is structured as follows. An overview of the relative navigation systems is presented in section 2. In section 3, we present the absolute navigation systems. Section 4 presents a discussion about the simulation together with its results. We conclude with a summary of the work, as well as possibilities for future research.

• *Sebastian Wehkamp is a Master student of Computing Science at the Rijksuniversiteit Groningen, E-mail: sebastianwehkamp@gmail.com.*

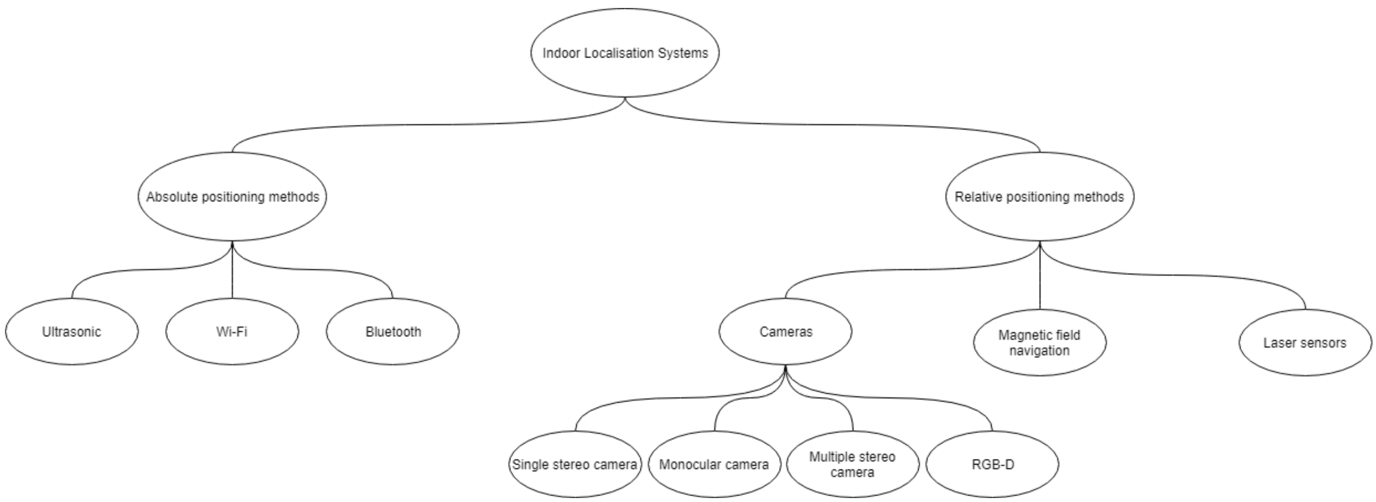
2 Relative positioning

This section describes several sensors used in relative positioning systems and how navigation is performed using these sensors. The first and most basic relative positioning sensor discussed in this section is the inertial measurement unit (IMU). The second relative positioning sensor discussed in this section is the camera. Camera configurations discussed in this section are a monocular camera , a single stereo camera, and multiple stereo cameras. The third relative positioning sensor discussed in this section is the laser scanner. The last relative positioning sensor discussed in this section is the magnetic field map.

2.1 Inertial measurement unit

Inertial measurement units (IMUs) are an often used relative positioning sensor. An IMU is an electronic device that measures forces acting on the agent and can include an accelerometers, gyroscopes, and sometimes magnetometers. Because of their small size, low cost, and low power consumption they are an often used sensor for indoor navigation [4, 14, 15]. Most studies use an IMU as the basis to perform simultaneous localisation and mapping (SLAM). This is a process where the agent tries to create a map of the environment (mapping) while trying to localise itself in the map (localisation). This is a complex problem because in order to localise oneself a map is needed and in order to map an environment one needs to know where you are localised. When using only the measurements of an IMU the agent is very sensitive for errors which keep increasing due to the next position estimate being dependent on the previous position estimate. This

Fig. 1: An overview of indoor positioning systems.



is why an IMU often serves as the basis for MAVs performing indoor SLAM and are often combined with other sensors.

2.2 Cameras

Cameras are used in a computer-vision-based approach which is very advantageous compared to other sensors because cameras are lightweight, have a low power consumption, and are relatively cheap [16]. The camera used depends on the size of the drone and the number of cameras used but most camera resolutions are in the range of 256p [14] to 1.6MP[4]. The frame rate which is used mostly depends on the computational resources available but most solutions aim for around 60fps. Some interesting features of a camera is that a camera has the capability to do 6-DoF pose estimation, 3D mapping, dynamic object tracking, and semantic labelling. However the algorithms to do this are computationally-intensive which might be a problem for MAVs. Because of the limited payload of MAVs this computing power is not always available [17]. Another disadvantage of cameras is that their performance is very sensitive to illumination so they do not perform well in dark indoor environments. There are various configurations in which cameras can be used for indoor navigation which will be discussed in this section [5, 4, 18].

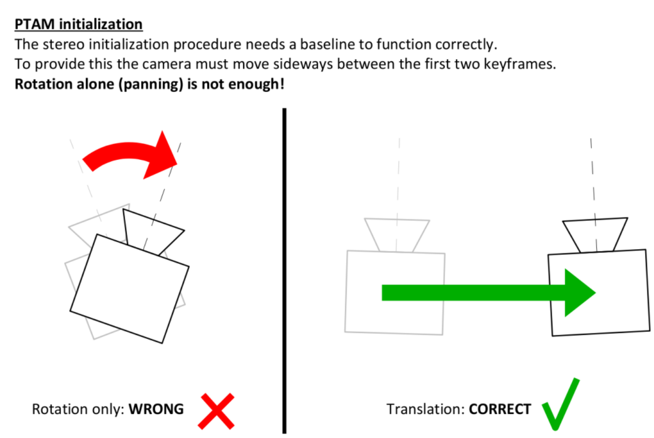
2.2.1 Monocular camera

Although the majority of the research done uses stereo cameras [17, 15, 4] approaches using a monocular camera are possible. The keypoint is to use translational movement to perceive points from different perspectives. This means that the SLAM algorithm must first be initialised with a specific translational movement an example of which is shown in Figure 2. The method can be summarised by the following points:

- Tracking and mapping run in separate parallel threads.
- Mapping is based on keyframes which are processed using batch techniques.
- The map is initialised as depicted in .
- New points are added using an epipolar search
- Large numbers of points are mapped.

The algorithm which implements this is called Parallel Tracking and Mapping (PTAM) [19]. Because the above is a challenging problem two limitations are imposed on the environment; it should be a mostly static and small environment. Besides the limitations imposed on the environment PTAM has another disadvantage, it has problems coping with pure rotations. This is because when a camera rotates around its centre the results will appear similar to that caused by a large translational movement. An MAV capable of autonomous flight which combines data from a forward facing monocular camera, downward facing camera, and an IMU is presented in [20, 21, 22]. The forward facing camera has a resolution of 640 x 360 with a field of view 92°which streams at 30fps. The downward facing camera has a field of view of 64°, a resolution of 320 x 240 and films at 60fps. The forward facing camera is used to perform PTAM while the downward facing camera is used to estimate the horizontal velocity. The MAV measures 53 x 52 cm and weights 420g. Because of the small size and low weight of the drone an external laptop is used to perform the computations. This MAV achieves an average positioning accuracy of 4.9 cm indoors and 18.0 cm outdoors and is capable of functioning with communication delays of up to 400 ms [20, 21, 22].

Fig. 2: How initialisation of PTAM should be performed[23].



2.2.2 Single stereo cameras

An MAV which mainly relies on a single stereo camera is described by [17] which combines the data from a stereo camera with the data from a downward facing optical flow sensor and an IMU. An optical flow sensor is a sensor which detects ground texture and visible features to determine the agents velocity. In this case the optical flow sensor is a custom built version which also uses an ultrasound sensor in order to estimate height of the drone.

The hardware of the MAV is a custom-built PIXHAWK quadrotor MAV with an Intel Core 2 Duo processor and 2 GB RAM onboard. The platform weights approximately 1.5 kg, and has a maximum payload of 600 g. The stereo camera consists of two Matrix Vision mvBlueFOX with an image resolution of 752 x 480 pixels. THe stereo baseline is set to either 80 or 120 mm.

First the metric velocity must be estimated using the optical flow sensor. A limitation of the optical flow sensor is that velocity measurements only work if the flight height of the MAV is at most 5m. This is because the optical flow sensor uses an ultrasound sensor to scale the results of the optical flow sensor and this ultrasound sensor has a maximum range of 5m. Another limitation is the maximum velocity of the MAV which is limited by:

$$v_{max} = \frac{whx}{f}$$

where w is the window size at x Hz, f is the focal length, and h is the height. In the experiments the MAV typically flies 1 m above the ground, and uses the following values: $w = 4$, $x = 250$, and $f = 666$. Therefore the maximum velocity that the optical flow sensor can register is 1.5 m/s.

The second step is to estimate the pose using visual odometry. This is done by computing poses with respect to a reference frame for subsequent frames. A new reference frame is selected when:

- The number of geometric inliers of the feature correspondences has to be above a threshold value.
- Either the Euclidean or angular distance measured between the existing reference frame and the current frame exceeds a threshold value.
- If no pose has been computed for a period of time

The usage of a reference frame has as an advantage that it is less susceptible to drift. This is because when you keep concatenating frame-to-frame relative pose estimates the chance for errors to accumulate increases.

The system described above is capable of detecting objects in front of the MAV using the front facing cameras. The pose estimates are combined with the metric velocity estimates using a simple Kalman filter. The MAV plans exploration strategies based on frontier-based exploration which works very well in cluttered environments with distinctive frontiers. However in open areas where distinctive frontiers are absent the algorithm under performs. [17] implemented a remedy solution based on wall-following which can optionally be selected as the substitute algorithm. Besides this limitation there is the limitation of a maximum flight height of 5m which might be solved by using a stronger ultrasound sensor [17]. In warehouses, where open areas are present, distinctive frontiers are absent. This limitation combined with a maximum flight height of 5m does not make this solution well suited for applications in large indoor environments inside warehouses.

2.2.3 Multiple stereo cameras

The limited field of view (FoV) of single stereo cameras is a problem when flying in constrained spaces where obstacles can surround MAVs like in indoor environments. Another area where a single stereo camera does not perform well is when fast yaw rotations occur. In this case, the camera registers fast image movements or even motion blur [15]. There are two common solutions which solve these limitations. The first solution is by having both a front facing and a downward facing camera [15]. The second solution is an omnidirectional configuration in which multiple stereo cameras are combined to provide complete omnidirectional vision [4].

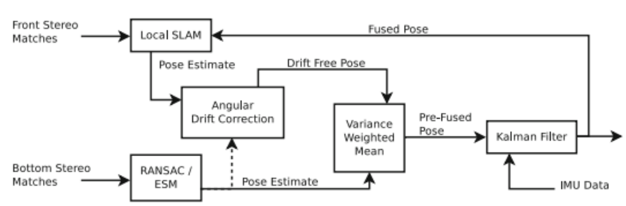
In the first solution the downward facing camera is used for detecting the ground distance and the orientation of the MAV towards the ground plane. Although a downward facing camera is better at detecting yaw rotations it only works when the MAV has reached a minimum altitude. The dissimilar strengths of downward and forward facing stereo cameras make it so that they complement each other well [15]. Their MAV uses a configuration in which they have one stereo camera oriented forwards and one stereo camera oriented downwards. The MAV is based on the PIXHAWHK platform and equipped with four USB cameras. Two cameras are facing forward with a baseline of 11 cm, while the remaining two cameras face downwards with a baseline of 5 cm. All of the cameras have a gray scale image sensor with a resolution of 640 x 480. The forward facing cameras operate at 30 Hz and the downward facing cameras operate at 15 Hz. The downward facing cameras operate at a lower frequency in order to lower the computational requirements. Although they operate at different frame rates the cameras are still synchronised with the downward facing camera skipping every other frame. The on-board computer consists of an Intel Core 2 Duo CPU running at 1.86 GHz [15].

The larger the number of cameras, the larger the computational needs. To solve this problem sparse stereo matching can be used which only provides a sparse set of matches [15, 24]. The algorithm is specifically designed for hardware platforms with limited processing power like MAVs. Because sparse stereo matching is much faster than any dense algorithm, the MAV can maintain a pose estimation rate of 30Hz on the on-board CPU. As processing time is very critical in this system instead of running feature re-detection at every scale level only those pixels that have a feature at the preceding level are re-evaluated. Because the computational requirements increase as the number of features increase or if, on the other hand, too few features are detected the performance will drop drastically a reasonable threshold is employed and an additional step of feature reduction on the left image is performed [25].

The forward facing camera is used for reduced stereo SLAM while the downward facing camera is used for ground plane detection and tracking. This makes it possible to obtain a full 6DoF pose estimate from each stereo camera. The stereo matching of the forward facing camera is done using the sparse stereo matching method described in the previous section. The stereo matching for the downward facing camera can be done more efficiently if it is assumed that the ground is flat. Under that assumption random sample consensus (RANSAC) based plane estimator can be used. The camera results are merged with the results from the IMU using an extended Kalman filter. The resulting pipeline can be found in Figure 3

The resulting MAV is capable of running the entire pipeline shown in Figure 3 on-board in real-time. Experiments have been done where the pipeline is executed with and without the data from the downward facing camera and the results show that using the downwards facing camera increase the pose estimation accuracy [15]. This MAV with dual stereo camera setup is capable of doing a 360°yaw rotation which is considered a very difficult manoeuvre for which the single stereo camera setup obtained very bad results.

Fig. 3: Processing pipeline using forward facing stereo camera and a downward facing stereo camera [15].



In the second solution [4] describe an MAV with an omnidirectional configuration. The hardware of the MAV consists of an Gigabyte GB-BXi7-4770R, Intel Core i7, 16Gb RAM, and a 480GB SSD. The cameras are six XIMEA MQ013MG-E2 with a resolution of 1.3 MP. The hexarotor design has a diameter of 1.24m with a total weight of 5.0 kg. The MAV uses six MK3644/24 motors with 14 propellers to generate thrust.

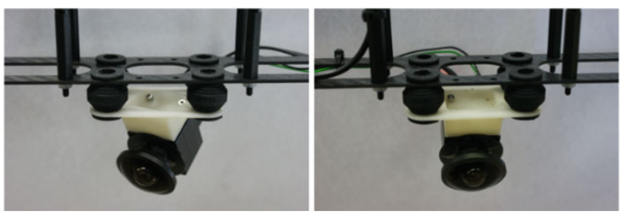
The MAV can switch between two camera configurations. The configuration seen in Figure 4a facilitates the usage of standard stereo methods while in the configuration in Figure 4b the cameras have a partial image overlap with the neighbouring cameras. This allows for truly omnidirectional vision methods. The MAV is capable of easily switching between the standard stereo configuration and the omnidirectional configuration as can be seen in Figure 5.

(a) three pairs of cameras.

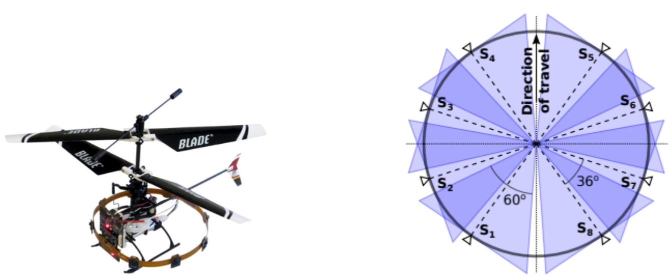
(b) omnidirectional configuration.

Fig. 4: Two different camera configurations to achieve omnidirectional vision [4].

Fig. 5: The left mount is for stereo configurations, the right mount is for omnidirectional camera configuration [4].



An omnidirectional setup is also possible in a very lightweight configuration of 30 grams [14]. [14] demonstrate an MAV, shown in Figure 6a, capable of autonomous speed control, centring, and heading stabilisation in a corridor. The MAV is based on a Blade mCX2 which is a small, low cost, coaxial helicopter which weights approximately 30 g and has a flight time in the order of 5 - 10 minutes. The processing is done by a low-power 32-bit Atmel micro controller (AT32UC3B1256). The MAV has a vision ring which consists of eight Centeye Paraya 64 x 64 px. The ring has a circumference of approximately 10 cm with a total weight of 2.5 g. Every camera has a FOV of 60°x 60°resulting in a combined FOV of 360°x 60°. The data from the vision ring was combined with the data from the optical flow sensor. The resulting MAV was capable of doing all sensing and all significant processing on-board at 10 Hz.



(a) The lightweight MAV.

(b) Camera configuration with FoV.

Fig. 6: The MAV together with its FoV [14].

The two different solutions both have different advantages and disadvantages. An omnidirectional configuration has as an advantage that it gives vision all around allowing the MAV to move in every direction without rotating. This however increases computational needs significantly when done using cameras with a high resolution of 1.3 MP. It also possible to accomplish such a configuration with a low amount of computational resources as shown by [14] however they use low resolution cameras of 64 x 64 px. If omnidirectional movement is not needed a configuration with a forward and a downward facing camera might be a solution which has lower computational requirements and increased accuracy over single stereo cameras.

2.2.4 RGB-D

RGB-D cameras provide an RGB image together with a per-pixel depth estimate. A variety of techniques can be used to produce these images like stereo cameras, laser sensors, or structured light stereo. Most of these techniques have been available to researchers for years but the recent application of these cameras in the entertainment industry resulted in a wide availability of low cost RGB-D sensors. An example of such a low cost RGB-D sensor is the Microsoft Kinect camera which can be bought for €25. This camera has a resolution of 640 x 480 px with a frame rate of 30 fps. A common approach in commodity RGB-D cameras is using structured light stereo. These cameras illuminate the scene with structured light pattern and they estimate depth using a single camera to detect the pattern. Because of this approach these cameras are able to estimate depth in areas with poor visual texture. However this limits their range to range of their projectors. The earlier versions of the Microsoft Kinect sensor use such an approach which has been used for indoor drone piloting research [18]. [18] present an approach for autonomous MAVs which performs reliable state estimates and creates a 3D map of its environment by using a stripped down Microsoft Kinect sensor which weighs 115 g.

This system enables autonomous MAV flight in unknown indoor environments. However in large open areas, the visible structures are often beyond the maximum range of the projector in the Kinect sensor. This results in that the system actually performs better in more cluttered, close quarter environments than it does in wide open areas. Another limitation is the amount of motion blur. In the paper it is assumed that the vehicle moves relatively slow, as when the vehicle flies faster the algorithms will need to handle larger amounts of motion blur and other effects caused by the rolling shutter in the Kinect cameras.

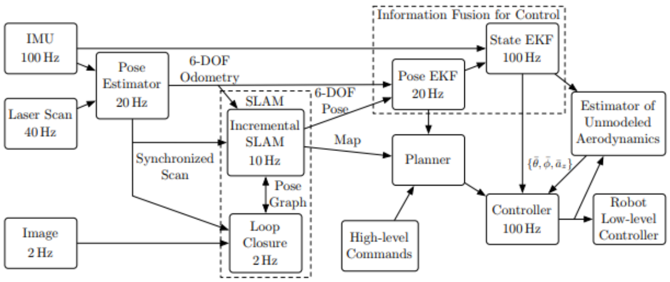
2.3 Laser sensors

A considerable amount of work has been done on using laser scanners in combination with autonomous MAVs. Compared to stereo cameras, laser scanners have a number of advantages. The main advantage is their accuracy which results in more accurate and reliable control, pose estimation, mapping, and navigation while using less computational resources. When combined with a mirror to redirect a few laser beams downwards, the laser can also be used to estimate the altitude of the MAV [15]. Compared to cameras they have a considerably higher weight and power consumption [15]. This can be seen when comparing for example an Hokuyo UST-20LX laser scanner and a Ximea mq013mg-e2 camera both used by [4]. The Hokuyo UST-20LX weights 130g and has a power consumption of 3.6W while the Ximea mq013mg-e2 weights 26 g and has a power consumption of 0.9W. The Hokuyo UST-20LX has a maximum detection distance of 60m and a scan speed of 25ms while having an accuracy of approximately 40mm.

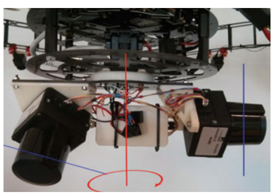
Typically, MAVs that rely on just laser scanners are unable to perform localisation in straight corridor indoor environments. This is because no unique structural features can be inferred from a laser scan which just shows two straight lines. As a result, these MAVs are unable to explore and navigate these environments [17].

In [26] a quadrotor MAV is presented which mainly relies on a laser scanner. This MAV is capable of doing autonomous path planning, and fully autonomous exploration of multi-floor indoor environments. The MAV does not require any external infrastructure to do the computation or any human interaction thus making the MAV fully autonomous. They use a combination of laser scans and IMU data for the pose estimator which is used for incremental SLAM. However when using just the results from the laser scanner and the IMU the MAV is not capable of reliably performing loop closure. To correct for this a monocular camera on which vision-based techniques are employed to enable robust loop closure which does not depend on the error of the pose estimator. This is done by matching SURF features of each incoming image with previous images via histogram voting. If there is a loop closure candidate, it is further verified using scan matching. The architecture diagram can be found in Figure 7.

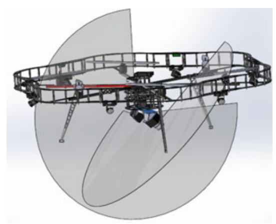
Fig. 7: Architecture diagram of MAV proposed by [26].



Laser scanners scan a single plane. To obtain a spherical FoV it is possible to place two laser scanners under an angle on a servo which makes the scanners rotate. An example of the lasers can be found in Figure 8a and in Figure 8b you can see the spherical FoV obtained from the laser scanners [4]. This drastically increases the weight and power consumption compared to a single laser scanner.



(a) Two laser scanners with rotation axis.



(b) FoV of the laser scanners.

Fig. 8: Image showing the two laser scanners together with the FoV [4].

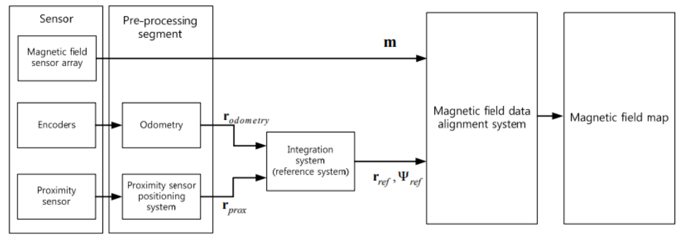
2.4 Magnetic field navigation

Structures like pillars, metal constructions, and large objects refract earth's natural magnetic field. Magnetic field navigation is based on the idea to detect these irregularities using a magnetometer and use them as clues for finding the agent's location. The approaches require mapping a given indoor environment beforehand which means measuring the magnitude and direction of the magnetic field at each point. The agent can find the most similar place in the magnetic field map to the one detected at a given point [27].

A way to determine the location of the agent in the map is by estimating where the agent is most likely to be. A way to do this is by performing a maximum likelihood estimation which is a method that uses the given observations to estimate the parameters of the model, in this case the possible locations. The higher the value, the more likely the agent is located at that location. The problem is that if many estimates have a high value the wrong location might be selected with a very large error. An alternative to maximum likelihood estimation is a particle filter which can select the optimal position even though many candidates have a high value [13]. To update the weights of the particles three magnetic field maps are used, a horizontal intensity map, a vertical intensity map, and a direction information map. The particles are propagated by a specific situation like the maximum speed. However if the robot is not in such a specific situation a particle filter is very inefficient and inaccurate. To solve this the magnetic map navigation can be combined with an inertial navigation system (INS). The INS is used to know the robots forward velocity and angular velocity in order to know in which direction the particles should be propagated. However, the INS has the problem of cumulative errors because every measurement depends on the previous one. Instead of using INS it is better to use odometry. These measurements for velocity do not depend on the previously measured velocity solving the cumulative error problem [13].

Although using odometry solved the cumulative error problem for velocity it does not for the estimated position. Therefore a way to calibrate this error is needed. This calibration can be done by using a proximity sensor of which the data is integrated with the odometry data. The final result can be found in Figure 9.

Fig. 9: Magnetic-field map building system [13].



Another approach to this problem is instead of determining the exact coordinates of the agent the goal is to identify the "room" in which the agent resides. Instead of creating a detailed magnetic map, a grid of magnetic field measurements, a signature for a given "room" is created. This signature is created by a random walk inside a given room. The frequency component of the magnetic signal is obtained using the Fourier transform of that signal. This method is independent of the exact path used when picking the magnetic signal making it robust [27].

A disadvantage for both methods is that the map or signature has to be created beforehand. When large object move in the environment the magnetic map will change as well making the current map obsolete. This means that every time the environment changes the map or signature will have to be updated as well. This makes these methods more useful in static environments.

3 Absolute Positioning

This section describes several techniques used in absolute positioning. Absolute positioning techniques use external devices to perform localisation of the agent. These external devices can use various techniques of which three common techniques are discussed in this section which are ultrasound, Bluetooth, and Wi-Fi. A common use case of absolute positioning systems is in combination with relative positioning methods to compensate for errors in the localisation of relative positioning methods. This is necessary because relative positioning often suffer from a cumulative error problem because the next location estimate depends on the previous location estimate.

3.1 Ultrasound

Absolute positioning methods using ultrasound transmission are widely used in robotics because it is simple, inexpensive, and accurate [6]. A disadvantage of ultrasound methods is that in general the range of ultrasound transmissions is limited to approximately 5m owing to the attenuation of the ultrasound signal [6].

There are various methods about how such a positioning method can be implemented but most of the methods depend on the time of flight (TOF) measurement. There is the bat system which consists of a single transmitter located on the robot and a matrix of receivers located in the indoor environment. An example of such a configuration is described by [28] where they create a matrix of receivers where the receivers were placed 1.2m apart in a volume of $75m^3$. They measured the TOF and were able to calculate the 3D position with a high accuracy of around 9 cm for 95% of the measurements. A drawback of this system however is the cost involved and scalability of the system. This is because there needs to be a fixed infrastructure placed on e.g. the ceiling with precise positioning. Because you can only localise a single bat at a time, the time space needs to be distributed efficiently [28, 29].

The second system is called the Cricket system which consists of a single receiver on the robot, and a distributed network of transmitters. To minimise signal collisions from the transmitters they broadcast at random times. There are a couple of benefits to this system. First, the scalability of the system will not be affected when more robots are added. Second, all information is received on the device which means that Cricket is not a centralised tracking system so no external calculations are necessary. This is also the disadvantage of the Cricket system, since all data is processed on the device the computational resources needed are high [30, 29]. A variation to this methods is a method which consists out of four transmitters which are located at the corners of a square space and a central controller. The controller sends a radio frequency (RF) trigger after which four precisely time ultrasound signals are emitted. The timing is very important in this scenario in order to avoid collisions [31].

The above two methods have a couple of disadvantages. The first disadvantage is that the transmitters cannot transmit at the same time. This causes the signals to interfere with each other making it difficult for the receiver to distinguish between them. The second problem is the sensitivity to noise from external sources. The first problem can be avoided by making sure the signal is transmitted one-at-a-time but the second problem is harder to address. Sounds occur often in indoor environments, sometimes without humans even noticing it. For the duration of such an occurrence, above solutions are prohibited from generating accurate position estimates. The Dolphin system overcomes these limitations by using broadband spread spectrum techniques. Transmitters emit a modulated signal simultaneously. The signal is modified by a predefined code instead of a burst pulse. The receiver collects the modulated signal and measures the TOFs using a cross correlation method. This makes the Dolphin method robust to noise and improves the range but requires many computations because of the cross correlation computation [6, 32]. The range in which methods using correlation possibly receive signal is approximately 30m compared to the 5m of normal ultrasound methods [6].

In recent studies research has been done on measuring time difference of flight (TDOF) instead of TOF. The use of TDOF instead of TOF removes the need for synchronisation between the transmitters and receivers because only the arrival times are measured. In a recent paper [6] a solution in which four transmitters are placed at one side of the space where the robot moves. In order to estimate the position the TOF is estimated. While the receiver is stationary it collects ultrasound signals at 200 kHz for 500 ms. From the four arrival times three TDOFs are calculated and modelled as hyperbolic curves. The position of the receiver is at the intersection of two hyperbolic curves. Note that this research has been done for 2D localisation. In order to apply it to MAVs a relatively easy solution would be to fix the height of the MAV at time of measurement [6].

3.2 Bluetooth

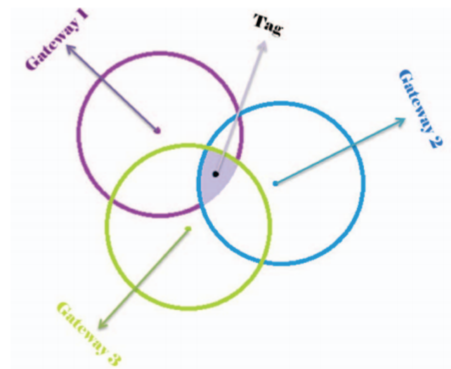
Since the release of Bluetooth Low Energy (BLE), an energy-efficient version of Bluetooth, Bluetooth localisation turns out to be a very practical localisation method. Another advantage of the newer version of Bluetooth is the improved range. Bluetooth 4.0 makes it possible to realise wide range indoor positioning due to the range which can be as long as 100 meters depending on the transmitting power and the environment. Because Bluetooth is available on inexpensive commodity devices this localisation method is very cost efficient.

A way to perform Bluetooth localisation is by using the Received Signal Strength Indicator (RSSI). This method determines a region in which the object is guaranteed to be found. There are two different versions of this, Low-precision Indoor Localisation (LIL) and High-precision Indoor Localisation (HIL). The difference is in the usage of the RSSI indicator resulting in a smaller localisation region for HIL compared to LIL [10].

LIL does not use the actual RSSI value, the only concern is whether the receiver receives the signal or not. Because the maximum range of the transmitters is known a circle can be drawn around each transmitter. Once one of the signals is received that circle is highlighted. If more than one signal is received the overlapping region will be highlighted, this can be seen in Figure 10 [10].

In contrast to LIL, HIL uses the RSSI value. This means that an extra training phase is necessary which is used to split the different power ranges into smaller bands. This results in a much smaller region. Both LIL and HIL in its current form only work in 2D environments due to the algorithm using 2D circles [10] .

Fig. 10: LIL algorithm where Tag is the robot and the gateways are the transmitters [10].



To use Bluetooth beacons in 3D another algorithm must be found. A solution to this problem is by placing four fixed position beacon nodes in the environment. It is important that the nodes are not in the same plane and they form a random sphere. The moving nodes (MAVs) must move inside the sphere. The positions of the moving nodes are calculated using triangulation. When the moving nodes are outside the randomly formed sphere a new sphere is formed which can include the moving nodes and the location can be calculated again using triangulation. The triangulation is done by using the RSSI value. This is also where the main issue lies. The RSSI value can vary because of various reasons making it difficult to calculate the value accurately because of interferences caused by Wi-Fi, environmental materials, or communication systems on the same frequency band. Therefore the main issue for using absolute positioning methods using Bluetooth beacons lies in reducing the error of the RSSI value [33].

3.3 Wi-Fi

Research done on using wireless networks for indoor positioning dates back until 1998 when Microsoft presented Microsoft Research Radar [34]. Microsoft Research Radar uses a combination of signal strength and dynamic changes such as temperature, the number of people present, and other environmental factors. Current state of the art wireless localisation solutions try to improve on these ideas.

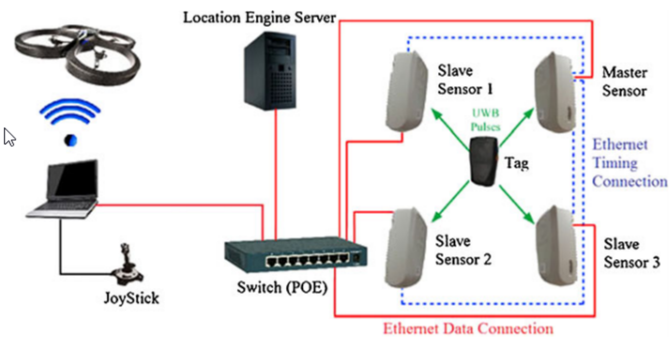
There are various standards but in particular IEEE 802.15.4 is very useful for its localisation features. It specifies two additional physical layers which can be used for ranging features. The first layer is the CSS layer which operates at 2.4 GHz ISM band. By itself this band does not support localisation features however the first 802.15.4 CSS chip (nanoLOC) developed by Nanotron has the ranging features included as a proprietary feature. Because of its immunity to the Doppler effect and long range it is a very good solution for MAVs moving at high speeds. A disadvantage is its lower bandwidth. The second layer is the Ultra-WideBand layer which operates below 1 GHz, between 3 and 5 GHz, and between 6 and 10 GHz. This band provides high data transfer, low power consumption, high spatial capacity of wireless data transmission and sophisticated usage of radio frequencies [35].

An example is given by [36] which uses the CSS layer combined with the technique described in a previous section about ultra sonic signals. Using the known coordinates of the beacons and the Time Difference of Arrival (TDOA), a hyperbolic curve is constructed to determine the position of the MAV. However this technique only works in 2D in its current form. This is why when performing localisation the altitude of the MAV is fixed. When communicating over the CSS layer symmetric double side two way ranging (SDS-TWR) is the key.

This means that every packet exchange is done twice, inverting the role of the two devices communicating in the second exchange [36]. This greatly reduces the impact of clock offset increasing the accuracy.

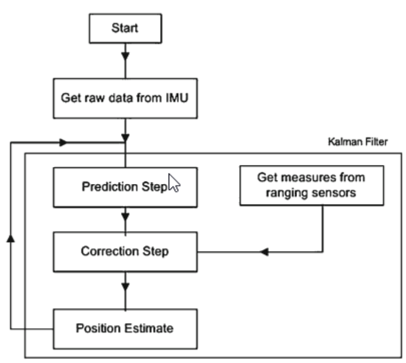
It is also possible to perform indoor localisation using wireless signals by using Ultra-WideBand (UWB) technology. A commodity MAV (Parrot AR) equipped with an UWB UbiSense Real-Time localisation system performs ranging measurements using two different methodologies, TDOA and Angle of Arrival (AoA). The combination of these two allows for a flexible system capable of performing localisation in three dimensions [35]. The MAV is equipped with a tag and sensors are placed throughout the environment. Each sensor is equipped with an array of four UWB antennas which are used for receiving the radio pulses of the tags. A diagram of the hardware architecture can be found in Figure 11 [35].

Fig. 11: A diagram of the hardware architecture [35].



The UWB localisation method is used to correct for the cumulative errors caused by the IMU unit. To correct for this the data from the IMU unit is fused with the data from the UWB system using an Extended Kalman Filter. The flow chart is depicted in Figure 12. [35] propose to further increase the accuracy of their localisation method by combining it with visual odometry methods [35].

Fig. 12: A flow chart of the sensor fusion algorithm [35].



Using this system an accuracy of $\leq$0.1 m can be achieved. They also note that their results reinforce the necessity to integrate additional sensors to obtain better accuracy and precision. They propose that for indoor environments the results could significantly improve when a Laser Range Finder is used [35].

4 Experiments

Due to the limited computational resources available on the MAV it is important to keep the resource usage of the various sensors and techniques in mind. This section will describe a simulation setup using existing modules in order to give an indication of the processing power and bandwidth used by various components. The experiments consist out of two important parts, a simulation engine and an operating system controlling the MAV. The simulation engine simulates sensor input which is directed to the operating system which processes these inputs, and outputs the desired motor velocities which are directed back to the simulation engine. Ideally, the simulation engine can be completely interchanged with an actual MAV without any changes to the operating system.

4.1 Experimental setup

A diagram showing the experimental setup is depicted in Figure 13. This configuration is very similar to the configuration in ”A modular Gazebo MAV Simulator Framework” [37]. Two experiments will be performed, one with a custom made MAV which has access to a laser scanner and a forward facing camera which can be used for manual navigation. The second experiment is based on a parrot AR drone 2.0. This MAV will use the monocular configuration discussed previously using a single forward facing camera. Because of the modularity of the experiments it is possible to identify the computational resources needed for every component.

4.1.1 Gazebo

The task of the simulation engine is to simulate all components which are on a physical MAV including a virtual environment for the MAV to fly in. I decided to use gazebo because it is open source and has good documentation about integration with the Robotics Operating System (ROS).

The gazebo configuration consists out of five parts as can be seen in Figure 13. The physics engine simulates the MAV dynamics and external influences acting on the MAV. Based on this simulation sensor outputs are generated which can be sent to ROS. Gazebo can receive control command in the controller interface which allows for controlling the MAV. At this point in time no absolute positioning methods are implemented in Gazebo however it should be possible to implement all of the absolute positioning methods discussed in this paper in Gazebo. Due to time constraints these are omitted in this experimental setup. To allow for easier debugging it is possible to give ROS access to the ground truth information about the location of the MAV.

4.1.2 Robotics Operating System (ROS)

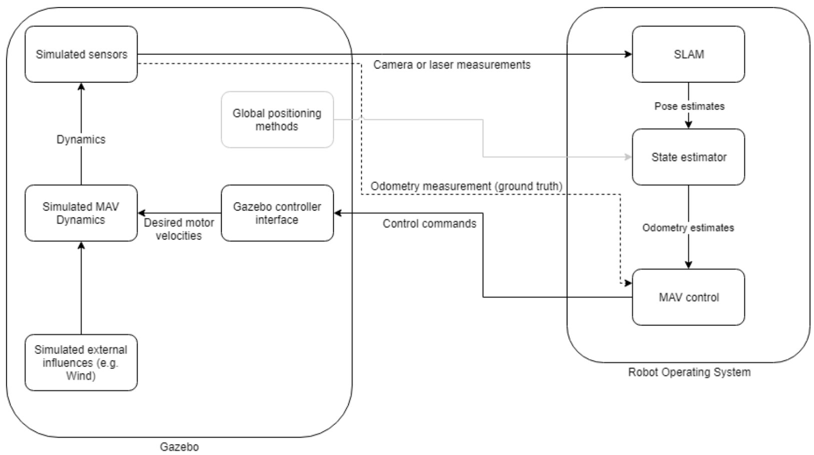
ROS is used because there is a large number of packages available which are easy to install and configure. There is good documentation available about the integration with Gazebo. There are various distributions of ROS where ROS Melodic is the newest LTS. However since this was only recently released most packages only work on the previous LTS which is ROS Kinetic. Therefore ROS Kinetic is used for this experiment. The ROS configuration consists out of 3 parts as shown in Figure 13. The SLAM component receives the simulated sensor inputs and simultaneously localises the MAV and maps the environment. This outputs to the state estimator which could combine the information of the local positioning methods with the absolute positioning methods. The output of the state estimator is sent to the MAV control components which generates MAV movements. Because of the modular design and an ROS package called *rqt* it is easy to identify resource heavy components.

4.2 Simulations

I have configured two different simulations which differ in the sensors used and in the SLAM method used. These packages are either available in the default Ubuntu repository or compiled from the respective Git repositories. The goal of these simulations is to see how resource intensive a specific version of SLAM is in terms of CPU and memory usage and how much bandwidth and processing a specific sensor needs.

The first simulation is called Hector Quadrotor [38] which is an ROS package used to model, control, and simulate quadrotor UAV systems. The package allows for Gazebo integration out of the box. There are two simulations available, one for indoor SLAM, and one for outdoor SLAM. The indoor simulation will be used for the experiment. The MAV itself uses a Hokuyo UTM-30LX laser scanner, there is also a front facing camera available which can be used for remote

Fig. 13: A diagram showing the structure of the experimental setup.



controlled flight. The environment it was test in is the default indoor environment for Hector Quadrotor. This environment consists of a large number of small rooms and hallways with some diagonal walls.

The second simulation is based on a Parrot AR drone 2.0 which is a small-scale, low-cost MAV with a monocular camera as the main sensor. This MAV combined with PTAM is capable of exploring unknown environments although the limitations discussed previously still apply i.e. it should be a small, mostly static environment. As discussed in Section 2, when using a single camera setup translational movements are needed in order to provide an accurate depth estimate. Because PTAM is used no markers, pre-made maps, known templates, or inertial sensors are necessary. All it requires is a monocular camera and stereo initialisation as previously shown in Figure 2. The simulation uses publicly available ROS packages based on [20, 21, 22]. This simulation is done in a custom made environment very similar to an actual warehouse, this means a lot of large open spaces and a large number of pathways.

4.3 Hardware

The experiments are done on a virtual machine (VM) with 8Gb of RAM and 4 CPU cores clocked at 2.2 GHz. Although this hardware might not be comparable with the hardware on an MAV it does give an indication of which components use a large amount of computational resources and bandwidth.

4.4 Results

In the first simulation, based on Hector Quadrotor, we are interested in how much CPU and RAM the hector mapping component uses. This component takes up 100% of a single core on average while using 100 megabytes (MB) of RAM. This result is achieved with a laser scanner running at 40Hz and pose estimates are updated at 20Hz. The laser scanner uses a bandwidth of 350 Kilobytes per second (KB/s) when running at 40Hz. A simplified diagram of the situation can be seen in Figure 14. Note that the results are maximum values found during a flight and not the average. The final map produced by the laser scanner with Hector SLAM is very accurate and the localisation is performed accurately.

In the second simulation, based on PTAM, we are interested in the CPU and RAM usage of two different nodes, drone state estimation which is the component performing PTAM, and drone autopilot which is the component that uses the PTAM data to control the MAV. The state estimation component uses 20% of a single core on average while using 65MB of RAM. The autopilot component uses 11% of a single core while using 15MB of RAM. A simplified diagram of the situation can be seen in Figure 15. When the initialisation is done properly the autopilot functions well in the simulated environment but the weakness for pure rotations is noticeable.

Fig. 14: A diagram showing the bandwidth and computational resources used for the various components in simulation 1.

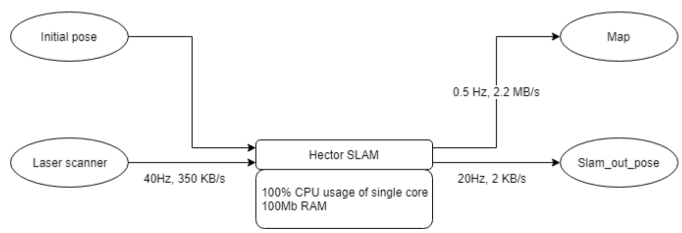
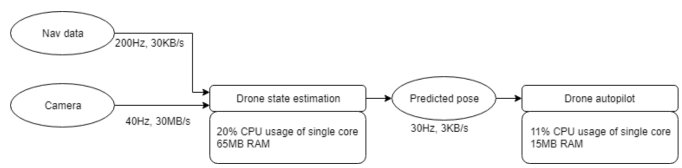


Fig. 15: A diagram showing the bandwidth and computational resources used for the various components in simulation 2.



4.5 Conclusion of experiment

It is clear that PTAM is optimised for devices with low computational resources. There is very little CPU and RAM usage so devices with a low amount of computational power should be able to run it without running into any problems. Hector SLAM is more CPU intensive but still only uses a single core to its fullest and the RAM usage is low. Although a direct comparison between ARM and x86 architectures is hard to do I would expect that a small device like a raspberry Pi is capable of running hector SLAM.

Another conclusion we can draw from above results is that, as expected, a laser scanner has a very low bandwidth, especially when compared to a camera. Compared to a camera with a bandwidth of 2.2MB/s a lot more data must be processed if a stereo camera is used. Higher bandwidth also means that processing on a ground station is harder to accomplish thus when using a camera since a faster connection is needed.

The simulations themselves work well and are very well suited for testing algorithms or to see an MAV functioning in a specific environment without actually having an MAV. The hector SLAM MAV only functions with manual flight in my situation, but it gives an accurate map of the environment. The PTAM MAV is capable of autopiloting through an environment as long as it is initialised properly and does not fly too high. In the PTAM simulation the expected weakness against rotation also became apparent.

5 CONCLUSION

Most of the research on indoor MAV piloting has been done on MAVs using relative positioning methods. They often use stereo cameras, laser sensors, or a combination of both. Where cameras have the advantage of being a low power, light, cost efficient solution while requiring more computational resources compared to a laser scanner which is a high power, heavy, high cost solution with superior accuracy and requiring less computations. Not much research has been done on magnetic field navigation combined with MAVs but it can be a way to improve the accuracy of indoor localisation techniques.

Most research on absolute positioning methods has been done on 2D localisation which is not sufficient for MAVs. Only in more recent papers research has been done on 3D localisation. The usage of absolute positioning methods looks very useful to correct for errors caused by relative positioning methods. They might remove the need of some relative positioning sensors in order to lower the power consumption of the MAV which can be useful in some scenarios. A disadvantage of most of these systems is configuring and placing the beacons. This is in some cases a large amount of work and very cost inefficient. Therefore Wi-Fi localisation looks especially promising since a lot of indoor environments have Wi-Fi beacons already installed. However you would still have to map the beacons because most algorithms require knowledge about the beacon placement.

Throughout the research some common factors can be identified. When using cameras for visual odometry it is important to use reference frames. A new reference frame is only selected when a condition is met. Another common factor throughout the available research is that the (extended) Kalman filter is the standard when combining data from multiple sensors to perform state estimation. In terms of hardware an often used starting point for the MAVs is the PIXHAWK platform. This can be combined with cameras with a resolution in the range of 64 x64 px to 1.3MP. An often seen brand of laser scanner is the Hokuyo which has various models depending on the use case.

The experimental configuration used in the experiments provide a modular framework in which Gazebo can almost be completely replaced by an actual MAV. It is possible to simulate realistic scenarios in order to see how the MAV would perform in the real world. Due to the modularity of the simulations it easy to identify how much computational resources a specific component uses. The conclusion is that both Hector SLAM and PTAM both do not require much computational resources and should be able to run on devices with a low amount of computational resources like a Raspberry Pi 3 with PTAM being more computationally efficient than Hector SLAM.

Not much research has been done on simulating absolute positioning methods which might be a future research topic. The main problem is simulating the behaviour of waves through certain objects and interference of materials. When these sensor values can be simulated, existing ROS packages can be used to interpret the results and existing algorithms can be executed. Simulating absolute positioning systems can be very interesting because purchasing and installing a large number of beacons used in absolute positioning methods purely for a test can be very expensive. When this is possible this might also stimulate the amount of research done on these methods.

Another interesting research we discovered too late is the research about is Large-Scale Direct Monocular Slam (LSD-SLAM) which looks like a modern version of PTAM. The algorithm is capable of performing SLAM using a single monocular camera and it might overcome some of the limitations of PTAM [39].

5.1 Use case conclusion

A couple of conclusions can be drawn from the research above with the use case of warehouse inventory in mind. PTAM is not a suitable solution for warehouses because it requires a mostly static environment which a warehouse is not. Structured light RBG-D cameras are also not a suitable solution because of the maximum range of these sensors. Almost all of the research done on this topic use an IMU so this would be the first sensor to install. Most of the research is done focusing on stereo cameras which can be used in various configurations. If omnidirectional movement is required an omnidirectional setup would be preferred if not then a forward and a downward facing stereo camera would be a good option. Another option would be to replace the downward facing camera with a laser sensor but this will increase the weight and power consumption of the drone. A magnetic field map does not seem like a good solution for warehouse inventory. The first obstacle is the large size of warehouses of which a map should be created, this will take a lot of time. The second problem is that in a warehouse large objects might move deprecating the map. Therefore magnetic field mapping does not seem like a good solution for warehouses. Of the absolute positioning methods ultrasound does not seem like the best option for warehouses due to the maximum range a lot of sound interferences in warehouses. Bluetooth might work although the range depends on the interferences of the metal structures which is hard to estimate. Therefore Wi-Fi looks like the best solution for a warehouse and an infrastructure might even be present already. Since the drones will not move at high speeds or at very long distances the UWB layer seems the most suitable option of the absolute positioning methods.

REFERENCES

[1] K. Kanistras, G. Martins, M. J. Rutherford, and K. Valavanis, "A survey of unmanned aerial vehicles (uavs) for traffic monitoring," pp. 221–234, 05 2013.

[2] Y. Naidoo, R. Stopforth, and G. Bright, "Development of an uav for search & rescue applications," *IEEE Africon 11*, 2011.

[3] M. Germen, "Alternative cityscape visualisation: Drone shooting as a new dimension in urban photography," *EVA London 2016 Electronic Visualisation and the Arts*, 2016.

[4] M. Beul, N. Krombach, M. Nieuwenhuisen, D. Droeschel, and S. Behnke, "Autonomous navigation in a warehouse with a cognitive micro aerial vehicle," *Studies in Computational Intelligence Robot Operating System (ROS)*, p. 487524, 2017.

[5] S. Grzonka, G. Grisetti, and W. Burgard, "Towards a navigation system for autonomous indoor flying," *2009 IEEE International Conference on Robotics and Automation*, 2009.

[6] B.-S. Cho, W.-J. Seo, W.-S. Moon, and K.-R. Baek, "Positioning of a mobile robot based on odometry and a new ultrasonic lps," *International Journal of Control, Automation and Systems*, vol. 11, no. 2, p. 333345, 2013.

[7] J. Villadangos, J. Urena, M. Mazo, A. Hernandez, F. Alvarez, J. Garcia, C. D. Marziani, and D. Alonso, "Improvement of ultrasonic beacon-based local position system using multi-access techniques," *IEEE International Workshop on Intelligent Signal Processing, 2005.*

[8] W.-J. Seo, B.-S. Cho, W.-S. Moon, and K.-R. Baek, "Position estimation using time difference of flight of the multi-coded ultrasonic," *Intelligent Robotics and Applications Lecture Notes in Computer Science*, p. 604609, 2011.

[9] S. S. Chawathe, "Beacon placement for indoor localization using bluetooth," *2008 11th International IEEE Conference on Intelligent Transportation Systems*, 2008.

[10] Y. Wang, Q. Ye, J. Cheng, and L. Wang, "Rssi-based bluetooth indoor localization," *2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, 2015.

[11] J. Biswas and M. Veloso, "Wifi localization and navigation for autonomous indoor mobile robots," *2010 IEEE International Conference on Robotics and Automation*, 2010.

[12] V. M. Olivera, J. M. C. Plaza, and O. S. Serrano, "Wifi localization methods for autonomous robots," *Robotica*, vol. 24, p. 455, Jun 2005.

[13] H.-S. Kim, W. Seo, and K.-R. Baek, "Indoor positioning system using magnetic field map navigation and an encoder system," *Sensors*, vol. 17, no. 3, p. 651, 2017.

[14] R. J. D. Moore, K. Dantu, G. L. Barrows, and R. Nagpal, "Autonomous mav guidance with a lightweight omnidirectional vision sensor," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[15] K. Schauwecker and A. Zell, "On-board dual-stereo-vision for the navigation of an autonomous mav," *Journal of Intelligent & Robotic Systems*, vol. 74, p. 116, Nov 2013.

[16] S. Xu, D. Honegger, M. Pollefeys, and L. Heng, "Real-time 3d navigation for autonomous vision-guided mavs," *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[17] L. Heng, D. Honegger, G. H. Lee, L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, “Autonomous visual mapping and exploration with a micro aerial vehicle,” *Journal of Field Robotics*, vol. 31, p. 654675, May 2014.

[18] A. S. Huang, A. Bachrach, P. Henry, M. Krainin, D. Maturana, D. Fox, and N. Roy, “Visual odometry and mapping for autonomous flight using an rgb-d camera,” *Springer Tracts in Advanced Robotics Robotics Research*, p. 235252, 2016.

[19] G. Klein and D. Murray, “Parallel tracking and mapping for small ar workspaces,” *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, 2007.

[20] J. Engel, J. Sturm, and D. Cremers, “Scale-aware navigation of a low-cost quadrocopter with a monocular camera,” *Robotics and Autonomous Systems*, vol. 62, no. 11, p. 16461656, 2014.

[21] J. Engel, J. Sturm, and D. Cremers, “Camera-based navigation of a low-cost quadrocopter,” *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[22] J. Engel, J. Sturm, and D. Cremers, “Accurate figure flying with a quadrocopter using onboard visual and inertial sensing,” *Proc. of the Workshop on Visual Control of Mobile Robots (ViCoMoR)*, vol. 320, 01 2012.

[23] D. M. Geargo Klein, “Parallel tracking and mapping for small ar workspaces initialisation,” 2014. [Online; accessed July 8, 2018], http://www.robots.ox.ac.uk/ gk/PTAM/ptam_initialization.png.

[24] K. Schauwecker, R. Klette, and A. Zell, “A new feature detector and stereo matching method for accurate high-performance sparse stereo matching,” *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012.

[25] K. Schauwecker, N. R. Ke, S. A. Scherer, and A. Zell, “Markerless visual control of a quad-rotor micro aerial vehicle by means of on-board stereo processing,” *Informatik aktuell Autonomous Mobile Systems 2012*, p. 1120, 2012.

[26] S. Shen, N. Michael, and V. Kumar, “Autonomous multi-floor indoor navigation with a computationally constrained mav,” *2011 IEEE International Conference on Robotics and Automation*, 2011.

[27] C. Galvn-Tejada, J. Garca-Vzquez, and R. Brena, “Magnetic field feature extraction and selection for indoor location estimation,” *Sensors*, vol. 14, no. 6, p. 1100111015, 2014.

[28] A. Ward, A. Jones, and A. Hopper, “A new location technique for the active office,” *IEEE Personal Communications*, vol. 4, no. 5, p. 4247, 1997.

[29] G. Deak, K. Curran, and J. Condell, “A survey of active and passive indoor localisation systems,” *Computer Communications*, vol. 35, no. 16, p. 19391954, 2012.

[30] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan, “The cricket location-support system,” *Proceedings of the 6th annual international conference on Mobile computing and networking - MobiCom 00*, 2000.

[31] C. Randell and H. Muller, “Low cost indoor positioning system,” *Ubicomp 2001: Ubiquitous Computing Lecture Notes in Computer Science*, p. 4248, 2001.

[32] M. Hazas and A. Ward, “A novel broadband ultrasonic location system,” *UbiComp 2002: Ubiquitous Computing Lecture Notes in Computer Science*, p. 264280, 2002.

[33] H. Park, J. Noh, and S. Cho, “Three-dimensional positioning system using bluetooth low-energy beacons,” *International Journal of Distributed Sensor Networks*, vol. 12, no. 10, p. 155014771667172, 2016.

[34] T. Liu, P. Bahl, and I. Chlamtac, “Mobility modeling, location tracking, and trajectory prediction in wireless atm networks,” *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 6, p. 922936, 1998.

[35] A. Benini, A. Mancini, and S. Longhi, “An imu/uwb/vision-based extended kalman filter for mini-uav localization in indoor environment using 802.15.4a wireless sensor network,” *Journal of Intelligent & Robotic Systems*, vol. 70, p. 461476, Oct 2012.

[36] J.-L. Rulln-Lara, S. Salazar, and R. Lozano, “Real-time localization of an uav using kalman filter and a wireless sensor network,” *Recent Developments in Unmanned Aircraft Systems*, p. 283293, 2011.

[37] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *RotorS—A Modular Gazebo MAV Simulator Framework*, pp. 595–625. Cham: Springer International Publishing, 2016.

[38] J. Meyer, A. Sendobry, S. Kohlbrecher, U. Klingauf, and O. von Stryk, “Comprehensive simulation of quadrotor uavs using ros and gazebo,” in *3rd Int. Conf. on Simulation, Modeling and Programming for Autonomous Robots (SIMPAR)*, p. to appear, 2012.

[39] J. Engel, T. Schps, and D. Cremers, “Lsd-slam: Large-scale direct monocular slam,” *Computer Vision ECCV 2014 Lecture Notes in Computer Science*, p. 834849, 2014.