Range based Algorithms for Precise Localization of Terrestrial Objects using a Drone☆

Francesco Betti Sorbelli[a], Sajal K. Das[b], Cristina M. Pinotti[c], Simone Silvestri[d]

[a]*Dept. of Computer Science and Math., University of Florence, Italy*
[b]*Dept. of Computer Science, Missouri University of Science and Technology*
[c]*Dept. of Computer Science and Math., University of Perugia, Italy*
[d]*Dept. of Computer Science, University of Kentucky*

Abstract

In this paper we propose two algorithms, called Dir and Omni, for precisely localizing terrestrial objects, or more simply sensors, using a drone. Dir is based on the observation that, by using directional antennas, it is possible to precisely localize terrestrial sensors just applying a single trilateration. We extend this approach to the case of a regular omnidirectional antenna and formulate the Omni algorithm. Both Dir and Omni plan a static path for the drone over the deployment area, which includes a set of waypoints where distance measurements between the drone and the sensors are taken. Differently from previously proposed best-effort approaches, our algorithms prove that a guaranteed precision can be achieved by considering a set of waypoints, for each sensor, that are at a distance above a certain threshold and that surround the sensor with a certain layout. We perform extensive simulations to validate the performance of our algorithms. Results show that both approaches provide a comparable localization precision, but Dir exhibits a shorter path compared to Omni, being able to exploit the directional antennas.

Keywords: Drones, terrestrial localization, localization precision, directional antenna, IR-UWB, omnidirectional antenna.

1. Introduction

More and more objects around us have sensors embedded with wireless communications: street lights, parking and gas meters, just to mention a few [1]. To bound their cost, each of them cannot be equipped with GPS units, but reliably and accurately determining their locations has important benefits for several applications such as routing protocols, intrusion detection, mission assignment and selective activation [2, 3].

Many existing localization algorithms for Wireless Sensor Network (WSN) in the literature require a large number of fixed anchor points, i.e., sensors whose positions are known a-priori [4]. The number of the anchor points and the cost of their deployment grow with the size of the

☆Part of this work has been accepted to the 19th International Conference on Distributed Computing and Networking. This manuscript has been accepted to Elsevier Pervasive and Mobile Computing (PMC). Digital Object Identifier no. 10.1016/j.pmcj.2018.05.007

Email addresses: francesco.bettisorbelli@unifi.it (Francesco Betti Sorbelli), sdas@mst.edu (Sajal K. Das), cristina.pinotti@unipg.it (Cristina M. Pinotti), silvestri@cs.uky.edu (Simone Silvestri)

deployment area. Moreover, the anchor points must be deployed in advance, making the use of anchor sensors unsuitable for emergency situations, unless anchors are equipped with expensive GPS units [5].

In order to decrease the setup costs of WSNs, we propose to replace fixed anchor sensors with a single mobile anchor, such as an Unmanned Aerial Vehicle (UAV). Recently, UAVs, also known as drones, have received increasing attention from research and industry community [6]. Drones can be used for military and civilian applications. As an example in civilian applications, a drone can be employed in rescue operations following disaster events, such as earthquakes, avalanches, shipwrecks, or nuclear disasters. Sensors can be randomly deployed in the disaster area and a drone can be used to be the contact point between the WSN and the external world.

However, localizing terrestrial sensors using a drone introduces novel and unique problems. As an example, for a precise range-based localization, the drone cannot take measurements directly above the sensor. In fact, in such setting a small imprecision on the line-of-sight (3D) distance between the drone and the sensor would translate into a large error when such distance is projected on the ground (2D distance).

In this paper we present two 2D range-based localization algorithms, called Dir and Omni, that use a drone as a mobile anchor. These algorithms make different assumptions of the hardware available at the drone to take distance measurements with the terrestrial sensors. Specifically, Dir assumes that the drone is equipped with directional antennas, which can adjust their beamwidth according to the precision required. Conversely, Omni assumes only a regular omnidirectional antenna. Both the solutions adopt the Impulse-Radio Ultra-Wide-Band (IR-UWB) technology [7] for the antennas. We assume that the drone is equipped with a GPS. During the localization process, at predefined positions called *waypoints*, the drone evaluates the distances between its current position and those sensors on the ground within its communication range using the IR-UWB technology. Subsequently, the sensors locally calculate their own positions through trilateration by using the measurements received from the drone. The waypoints are defined by a static path over the deployment area. Dir and Omni guarantee the sensor localization precision required by the final user of the sensor network.

The main contributions of this paper are the following:

- We formally provide an expression for the localization error given the minimum distance and the geometrical locations of a sensor and its waypoints.
- We introduce a localization algorithm, called Dir, which assumes the drone is equipped with directional antennas. Thanks to this assumption, we show that Dir obtains the required localization precision using just a single trilateration.
- Since directional antennas may not be available or economically convenient for off-the-shelf drones, we propose Omni that is able to guarantee the required localization precision using an omnidirectional antenna, at the cost of an additional trilateration for each sensor and a longer static path.
- We show that both Dir and Omni have the ability to control the localization error just setting the minimum distance between a sensor and its waypoints. At the best of our knowledge, they are the first algorithms in literature that can be tailored for ensuring the precision needed by the final user of the WSN.

- We also validate DIR and OMNI through extensive simulations. Results show that our algorithms scale well with respect to the network size, and are efficient in terms the length of the path traversed by the drone. Our algorithms do not require any pre-processing phase or expensive computation, such as solving instances of the Traveling Salesman Problem.

The rest of paper is organized as follows. Sec. 2 revises the current literature on localization algorithms using mobile anchor nodes. Sec. 3 introduces the model and gives an expression for the localization error as a function of the ground error and the geometry of the waypoints. In Sec. 4, we present DIR, while in Sec. 5 we present OMNI. The experimental results are discussed in Sec. 6, while Sec. 7 concludes the paper.

2. Related Works

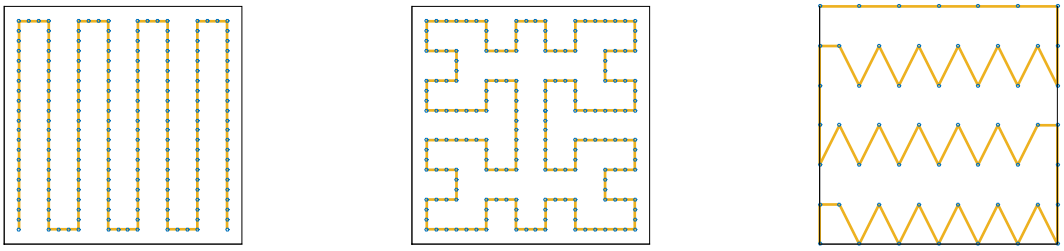
In recent years, with the advent of autonomous vehicles, the interest of researchers in rovers, drones, and other autonomous vehicles has increased [8]. Depending on the technique used by the mobile anchor to estimate the sensors' positions, localization algorithms can be categorized in *range-free* or *range-based*. In the former case, the position estimation is performed without using any type of ranging measurements [9, 10]. In the latter, the estimations are instead inferred by exploiting several properties of the communication signals, such as the Received Signal Strength Indicator (RSSI), the Time of Arrival (ToA), the Time Difference of Arrival (TDoA), and the Angle of Arrival (AoA) [11].

Given the additional degree of freedom of mobile anchors, compared to static anchors, it is necessary to define a path in the deployment area which is followed by the anchor and used for localizing the sensors. At the best of our knowledge, no previous algorithm in literature, except LOCALIZERBEE in [12], studies the path planning problem for localizing sensors using a drone. So except for LOCALIZERBEE, all the localization algorithms that we survey are for mobile anchors, such as patrols, moving on the ground. In general, the requirements for path planning are essentially three: (i) the path should be in range of as many sensors as possible, in order to increase the number of sensors for which a precise localization is provided, (ii) it should provide at least three non-collinear waypoints for each sensor in the 2D space, in order to use triangulation effectively, and (iii) it should be as short as possible in order to reduce the energy consumption of the anchor sensor.

Path planning schemes can be classified into two sub-categories, *dynamic* and *static*. In the former category, the path trajectory is decided during the localization process. This is preferable, for example, when sensors are expected not to be uniformly deployed over the area, and the path can be adapted according to their actual distribution.

A dynamic localization technique is presented in [13] for obstacle-free environments. The approach progressively changes the status of the sensors in the network from unknown sensors to reference sensors, that is, sensors that have been localized and can act as anchors for the unknown sensors. The algorithm consists of three sub-phases: reference movement phase, sensor localization phase, movement path decision phase. Based on the position of the more recently localized sensors, the mobile anchor sensor decide its movement trajectory.

In [14], instead, a dynamic path which has the ability to face and detour obstacles, following a snake-like pattern is presented. When the mobile anchor sensor faces the obstacle, it changes its direction and stores all the information for future movements.



(a) Scan (b) Hilbert (c) Lmat

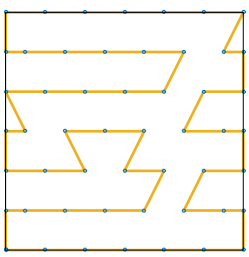
Figure 1: Different movement trajectories

The static localization techniques are preferable when sensors are almost uniformly scattered over the deployment area, and it is important to localize all of them. In the literature, for terrestrial mobile anchors, several 2D static path planning schemes, such as Scan, and Hilbert [11], have been studied. According to Scan (see Fig. 1(a)), the mobile anchor sensor follows a path formed by vertical straight lines interconnected by horizontal lines. The main drawback of this approach is that it provides a large amount of collinear anchor points. Collinearity can be reduced by increasing the changes of directions in the trajectory, as for the path Hilbert illustrated in Fig. 1(b). However, the changes of directions significantly increase the path length.

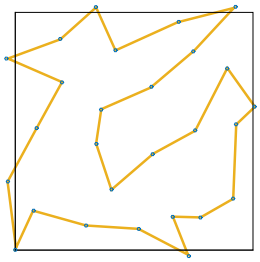
The authors of [15] develop Lmat (see Fig. 1(c)), one of the best state of art static schemes addressing to the collinearity issue. The path generated by Lmat logically tessellates the deployment area by equilateral triangles so as each terrestrial sensor falls inside a triangle. The vertices of the triangle where the sensor resides are used to trilaterate the sensor position, thus completely solving the collinearity issue. However, this is realized through often occurring sixty degree turns, which are more challenging for general purpose mobile vehicles than straight paths [8]. Overall, Lmat decreases the path length, but increases the level of difficulty of the path.

All the approaches described above make use of terrestrial mobile anchors and we refer the reader to [11] for an exhaustive survey. As the survey shows, the accuracy of such techniques is in the orders of meters (often as a percentage of the communication range), while the algorithms proposed in this paper are in the orders of few decimeters.

To the best of our knowledge, the first localization algorithm that uses a drone appears in [12], but the goal there is to securely localize all sensors in a generic deployment area. Three path planning algorithms, LocalizerBee, VerifierBee, and PreciseVerifierBee are proposed that allow a drone to respectively localize, verify, and verify with a guaranteed precision in a secure manner. LocalizerBee uses a static path algorithm and it can be considered an improvement of Lmat. The static path is built in two different phases: waypoint grid construction and waypoint reordering. In the first step, the sensing area is logically tessellated by a set of waypoints forming isosceles triangles, whose sides are resized with respect to the size of deployment area. In the second step, a Traveling Salesman Problem (TSP) is solved in order to connect all the vertices of the isosceles triangles. The resulting path, which is more similar to Scan than Lmat (see Fig. 2(a)), is definitely shorter than that of Lmat. This result is achieve at the cost of a high computational step (i.e., invoking TSP solver), which has to run off-line and using a different hardware platform. The final static path of LocalizerBee, due to the smallest number of turns, it is also simpler to execute for a drone than that of Lmat. Moreover, like Lmat, LocalizerBee does not ensure any localization precision. Although the precision of the localization is studied in LocalizerBee, it is



(a) LocalizerBee



(b) PreciseVerifierBee

Figure 2: Example of LocalizerBee and PreciseVerifierBee paths for drones

not possible to tailor LocalizerBee so as it guarantees the localization precision required by the final user. The second path planning approach proposed in [12] is VerifierBee, which focuses on finding a path of minimum length for verifiable multilateration. Differently from our problem, the goal here is to securely verify the positions of the sensors, even under potential attacks aiming at spoofing the sensors' positions. As a result, VerifierBee requires the previous knowledge of the positions to be verified. The algorithm starts selecting three waypoints that form an equilateral triangle around each terrestrial sensor, and solves an instance of the TSP problem based on such waypoints. Then, to improve the path length, two operations are allowed on the waypoints: pruning and reordering. These are repeated until significant improvement is achieved on the length of the path computed by applying the TSP solver on the new set of waypoints. Finally, the third path planning approach in [12] called PreciseVerifierBee, follows the same paradigm of VerifierBee, but it also guarantees a precise localization by carefully selecting the vertices around each terrestrial sensor. Both PreciseVerifierBee and VerifierBee compute the drone's trajectory ad-hoc for each set of sensors (see Fig. 2(b)), by assuming a previous knowledge of the sensors' positions, which was the main reason to design LocalizerBee. Therefore, PreciseVerifierBee cannot be considered a precise localization technique.

To the best of our knowledge, Dir and Omni are the first two algorithms that use drone to localize terrestrial sensors with guaranteed precision. A preliminary version of Dir algorithm was presented in [16], while a preliminary version of Omni was originally presented in [17]. In the current paper, we describe these two algorithms in a novel way under the same framework. Namely, the logical tessellation for Omni discussed in Sec. 5 becomes relevant by itself because it shows a way of emulating the directional antennas in Dir using omni-directional antennas. Moreover, under the same framework, it is easier to make a comparative analysis of the performance trade-off between these two algorithms. To be precise, Omni achieves a slightly better precision whereas Dir incurs a lower computational cost.

3. Background

We consider a network of n sensors deployed in a rectangular area Q of size $Q_x \times Q_y$. Without loss of generality, we assume $Q_y \leq Q_x$. From now on, sensor P is short form for the sensor that resides at a point P in Q. Since, in principle any point P in Q is candidate to contain a sensor, with a little abuse of notation, we denote P indistinctly as the point P or the sensor P. Moreover, we use the notation $\overline{PP'}$ to denote the ground distance between two waypoints/sensors P and P'.

The drone acts as a mobile anchor that flies at a fixed altitude h [18]. We assume that the drone

measures its altitude with negligible error, for example, by using the Differential GPS technique [19]. To range the sensors on the ground and to achieve a high precision, we adopt the two-way ranging application of the Impulse-Radio Ultra-Wide-Band (IR-UWB) technology. In fact, IR-UWB guarantees a very high measurement precision of the order of 10 cm [20, 7], but other technologies could be used as well. As it will be clear in Sec. 3.4, our results still hold, but they will be scaled up according to the instrumental precision. For example, WiFi guarantees an instrumental precision of 7-10 m, and Bluetooth of 15 m [21, 22].

3.1. Static Path Definitions

To perform the localization mission, our algorithms need to calculate the static path that the drone follows during the localization process. The two algorithms DIR and OMNI derive two slightly different static paths, called Π_D and Π_O, respectively. Both the paths are formed by vertical and horizontal scans, like the simple trajectory SCAN. The vertical scan are parallel to the short side Q_y of Q.

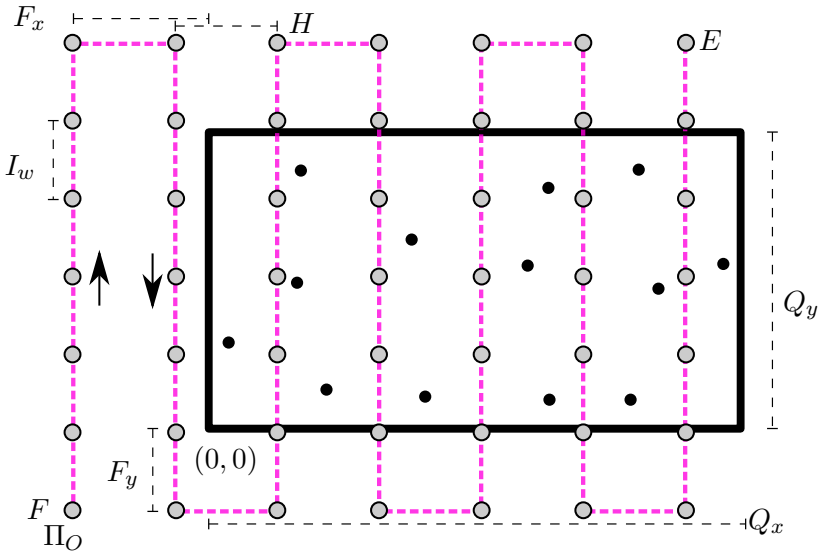


Figure 3: A generic static path (the segment $\overline{EF}$ is not sketched). The vertical scans are at reciprocal distance H.

The static paths can be described in terms of few parameters: the starting point F, the ending point E, the frontiers F_x and F_y, and the inter-scan H. The drone's mission starts outside the deployment area at the point $F = (-F_x, -F_y)$. Each vertical scan has length $Q_y + 2F_y$. Any two consecutive vertical scans are at distance H and they are connected by an horizontal segment. The last vertical scan finishes at the point $E = (x_E, y_E)$, where y_E can be either $-F_y$ or $Q_y + F_y$ depending on the parity of the number of vertical scans. The last vertical scan path finishes at E, and then the drone goes back to its starting point F following the direct line between E and F. In this way, the starting and the finishing point coincide, as it is usually required for drone's path.

3.2. Measurement Definitions

During the mission, the drone takes measurements at pre-established points of the path. The projections of such points on the ground are called *waypoints*. The waypoints reside only on the vertical scans. The *inter-waypoint distance* between two consecutive waypoints is denoted by I_w.

We assume that the drone, using the IR-UWB antenna, measures its distance from the sensors from the round-trip time of messages exchanged with them. To take a measurement, the drone acts as follows. At each waypoint w_i, the drone sends a beacon with a unique identifier which depends on the coordinates (x_{w_i}, y_{w_i}) of the waypoint and the current timestamp. Each sensor on the ground that can hear a beacon replies to the drone with an ack message that contains its ID, the current timestamp, and the identifier of the beacon received. The antenna (for simplicity, we say the drone) computes the distance between itself and the sensor using the round-trip time of the beacon message and then sends a message with the computed distance to the sensor. Once a sensor has collected three measurements, it can locally apply the trilateration algorithm tocalculate its position P exploiting the distance measurements received by the drone. Hence, given three ground measures, the estimated position of P is the point (x_P, y_P) that minimizes the sum of the least squares, that is:

$$\begin{aligned} &\min \quad \delta_1^2 + \delta_2^2 + \delta_3^2 \\ \text{s.t.} \quad &\sqrt{(x_{w_i} - x_P)^2 + (y_{w_i} - y_P)^2} + \delta_i = \overline{w_i P} \text{ , for } \quad i = 1, 2, 3. \end{aligned} \tag{1}$$

3.3. Distance Definitions

The drone and the sensor have communication ranges, r_{drone} and r_{sensor}, respectively. Since a message can be exchanged between the drone and the sensor only if they can hear each other, throughout this paper, the communication range will be $r = \min\{r_{\text{drone}}, r_{\text{sensor}}\}$.

The drone measures the (3D) *slant distance* s, which is defined as the line-of-sight between itself and the measured sensor. Clearly, $s \leq r$. By simple geometric argument, since we assume negligible error in altitude, it is easy to see that any *ground* measured distance d satisfies:

$$d = \sqrt{s^2 - h^2} \leq \sqrt{r^2 - h^2} = d_{\max} \tag{2}$$

In the next subsection, we find the expression for the localization error as a function of the ground distance, the slant precision, and the geometry of the waypoints.

3.4. Precision Definitions

Slant distances are affected by errors that depend on the adopted technology, i.e., IR-UWB in our case [20]. From now on, the *slant precision* or *instrumental precision* ϵ_s denotes the maximum error in the measurements. The trilateration algorithm works on the ground distances, thus the *ground error* $e_d(P)$ for a point P on the ground is [12, 23]:

$$e_d(P) = e_s \cdot \frac{1}{\cos(\alpha)} = e_s \cdot \sqrt{1 + \frac{h^2}{d^2}} \tag{3}$$

where α is the angle of incidence of the slant distance to the ground, d is the actual distance on the ground between the drone and the sensor to be localized, h is the current altitude, and e_s the actual measured slant distance error (see Fig. 4).

From Eq. (3), the ground error increases when the ground distance decreases. Thus, in order to achieve a *high precision on the ground* ϵ_d, or equivalently, *to minimize the maximum ground error*, we force a constraint on the ground distance. Precisely:

Fact 1. *The drone retains only the measured ground distances greater than or equal to* d_{min}.

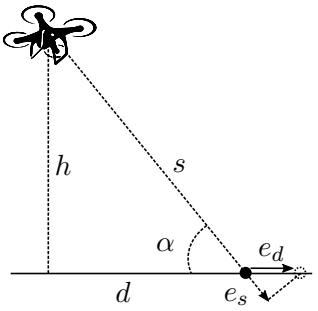


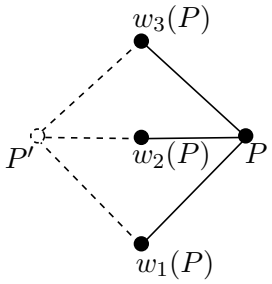
Figure 4: Relationship between slant error and ground error

All our algorithms force any sensor P to be measured only by waypoints which are at distance at least $d_{\min}$ from P. As we will see, our algorithms select the value of $d_{\min}$ which guarantees the user-defined precision. Since from Eq. (3) the slant error is maximum when the distance on the ground is minimum, fixed $d_{\min}$, the maximum possible ground error is:

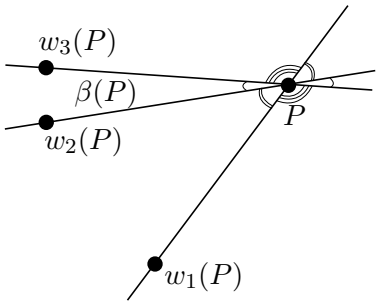
$$\epsilon_d = \max_{P \in Q} e_d(P) = \epsilon_s \cdot \sqrt{1 + \frac{h^2}{{d_{\min}}^2}} \tag{4}$$

As a consequence, a peculiarity of all our algorithms is to leave a no-measurement area of width $d_{\min}$ all around each sensor. Clearly, the ground precision is higher (or, equivalently, the error in the ground measure is smaller) for larger values of $d_{\min}$.

Moreover to minimize the *localization error* $e_L(P)$ due to the trilateration, the three waypoints from which a sensor is measured cannot be collinear neither among them nor collinear with the sensor. In fact, if the waypoints are collinear among them, the trilateration algorithm cannot distinguish between the real position P of the sensor and the mirror image P' of P (see Fig. 5(a)).



(a) Among the ranging waypoints



(b) With the sensor to be localized

Figure 5: Collinearity problems

With regard to the collinearity with the sensor (see Fig. 5(b)), as proved in [12, 23], the localization error is expressed as:

$$e_L(P) = \frac{\epsilon_d}{\sin\left(\frac{\beta(P)}{2}\right)} \tag{5}$$

where $\beta(P)$ is the minimum angle formed by the lines passing through each waypoint and the

sensor P. Considering the worst case, from the construction in Fig. 6 studied in [12, 23], the *localization precision* ϵ_L is expressed as:

$$\epsilon_L = \max_{P \in Q} e_L(P) = \frac{\epsilon_d}{\sin\left(\frac{\beta_{\min}}{2}\right)} \tag{6}$$

where $\beta_{\min} = \min_{P \in Q} \beta(P)$ is the minimum angle obtained during the localization. If the waypoints are collinear with the sensor (see Fig. 5(b)), $\beta_{\min}$ tends to 0 and the localization error becomes very large and thus the position very imprecise. Since the localization error decreases when $\beta_{\min}$ increases, it holds:

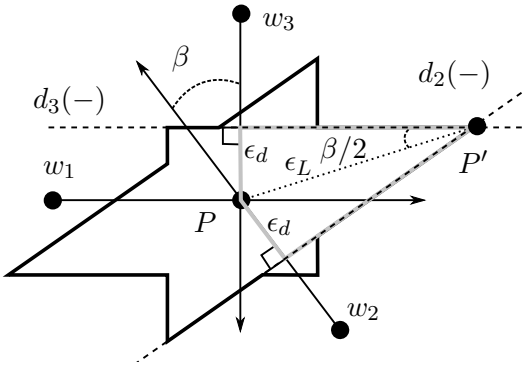


Figure 6: Star shape during trilateration

Fact 2. *The best geometry in P occurs when the turn angle in P is divided by the lines passing through the waypoints in six equal angles, and thus when $\beta_{min} = \frac{\pi}{3}$.*

By Fact 2, the minimum localization error is $\epsilon_L = \frac{\epsilon_d}{\sin(\pi/3)} = \frac{2}{\sqrt{3}}\epsilon_d$. Unfortunately, it is not possible to achieve such a minimum error for all the points of the deployment area.

For both DIR and OMNI algorithms, however, we find which is the minimum angle $\beta_{\min} = \min_{P \in Q} \beta(P)$ ensured by the geometry of the three selected waypoints. So, we know in advance how far is $\beta_{\min}$ in DIR and OMNI from the optimal value $\frac{\pi}{3}$.

To summarize, the aim of our algorithms is to obtain an ϵ_L *precise localization* for each point P in Q, where the localization maximum error ϵ_L is given by Eq. (6) as a function of the parameters $d_{\min}$ and $\beta_{\min}$. Formally, we define the precise localization as follows:

Definition 1. *The sensor P is ϵ_L-precisely localized, where ϵ_L is given by Eq. (6), if the drone chooses three ranging waypoints $w_1(P)$, $w_2(P)$ and $w_3(P)$ for P such that they satisfy the following constraints:*

1. *d_{max}: which controls the reachability for each point P in Q: $\overline{w_i(P)P} \leq d_{max}$ for $i = 1, 2, 3$;*
2. *d_{min}: which controls the ground precision ϵ_d for P in Q: $\overline{w_i(P)P} \geq d_{min}$ for $i = 1, 2, 3$;*
3. *β_{min}: which controls the collinearity of the drone with P: $\beta(P) \geq \beta_{min}$*
4. *non-linearity: which controls the collinearity among waypoints: $w_1(P)$, $w_2(P)$, and $w_3(P)$ cannot belong to the same straight line.*

Table 1: Summary of Notation

symbol	description	unit
w	waypoint – (x_w, y_w) drone's position	(m, m)
P	point (x_P, y_P)	(m, m)
α	angle of incidence	rad/deg
h	drone's altitude	m
d	ground (2D) distance	m
s	slant (3D) distance	m
r	communication range	m
ϵ_s	global instrumental slant precision (IR-UWB: 0.10 m [20])	m
$e_d(P)$	ground error for a point P	m
ϵ_d	highest ground precision (max ground error)	m
$e_L(P)$	localization error for a point P	m
ϵ_L	highest localization precision (max localization error)	m

Tab. 1 summarizes our notations. From now on we will neglect the units of measurement. In the next two sections, we presents our new algorithms Dir and Omni, and we theoretically evaluate their precision.

4. The DIR Algorithm

In this section we describe the algorithm Dir, which assumes the drone to be equipped with directional antennas. The section is structured as follows. In Sec. 4.1 we describe the hardware assumptions for this algorithm. In Sec. 4.2 we detail the calculation of the static path, while in Sec. 4.3 we formalize the localization process and formally evaluate its precision.

4.1. Model of Directional Antennas

The drone is equipped with six directional antennas. Each antenna is assumed to transmit the beacon in a circular sector, centered at the antenna position, of radius r, beamwidth 2θ and orientation ψ. Fig. 7 depicts a sector centered at waypoint w with direction ψ. In the following, a sector will be uniquely identified by its center and its orientation because we assume that all the sectors have the same radius and the same beamwidth.

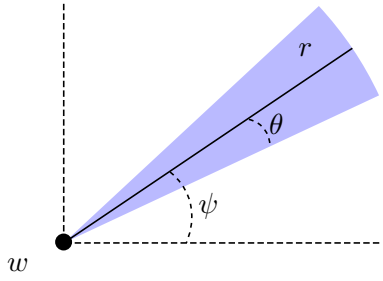


Figure 7: The sector parameters

In our solution the directional antennas cover six different sectors, as illustrated in Fig. 8. The drone can send at the same time the beacons in all six orientations using an array of directional antennas [24]. Note that the sensors on the ground are equipped with omnidirectional antennas.

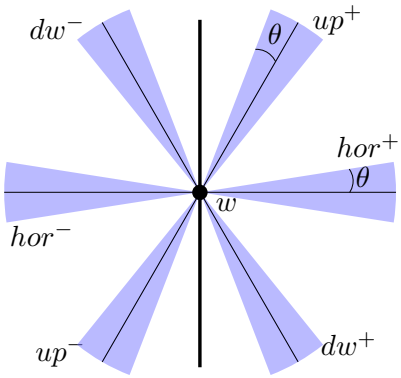


Figure 8: Possible antenna orientation

From now on, we express each orientation ψ as a pair: *type* (*up*, *dw*, *hor*) and *polarity* (+, −), as reported in Tab. 2. Each sensor saves in its local register R the first beacon that it receives for each orientation. The matching between the register positions and the orientations is also reported.

4.2. Static Path

The static path Π_D is depicted as a dashed line in Fig. 9, where the gray dots represent the waypoints, while the black dots are the sensors inside Q. Π_D is uniquely determined posing $F = (-F_x, -F_y)$, where $F_x = d_{\min}/2$ and $F_y = 0$, and $E = (E_x, E_y)$, where $E_x = Q_x + F_x$ and E_y can be either 0 or Q_y depending on the parity of the number of vertical scans. It is important to note that Dir does not exit from the bottom and top border of Q.

From each vertical scan σ, the points of Q that we can precisely measure using any type of orientation are those at distance at least $d_{\min}/2$ and at most $(d_{\max} - 2I_w)/2$ from the scan σ. In fact, although the antennas of type *hor* can precisely measure the points at distance from $d_{\min}$ up to $d_{\max}$ from σ, the most stringent limits for the precise measurements come from the antennas of type *up* and *dw*. Therefore, any two consecutive vertical scans are fixed at distance no greater than at distance greater than $(d_{\max} - d_{\min} - 2I_w)/2$. Having fixed the first and the last scan respectively at $-F_x$ and $Q_x + F_x$, the length of H is fixed in order to evenly distribute $Q_x + 2F_x$ in stripes of width as tight as possible to the maximum value. In this way, the whole area is covered without wasting path length. Finally, observe that the left and right stripes of Q of width $d_{\min}/2$ adjacent to each vertical scan σ are not measured by σ to satisfy the $d_{\min}$ constraint. The left and right stripes adjacent to σ are then measured by the scan that precedes σ and the scan that follows σ.

Table 2: Antenna orientation and register R

$R[d]$	ψ	type	description	polarity
0	0	*hor*	horizontal	+
1	$(1/3)\pi$	*up*	up	+
2	$(2/3)\pi$	*dw*	down	−
3	π	*hor*	horizontal	−
4	$(4/3)\pi$	*up*	up	−
5	$(5/3)\pi$	*dw*	down	+

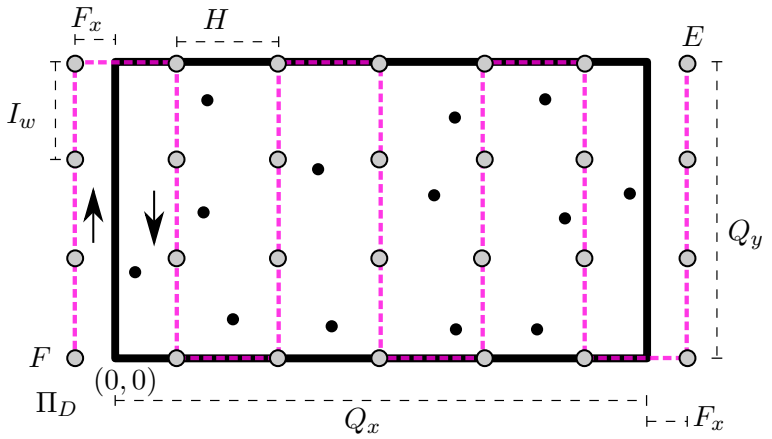


Figure 9: The static path Π_D of DIR (the segment $\overline{EF}$ is not sketched)

4.3. Localization Process

Algorithms 1 and 2 describe the drone's and sensors' behavior during the localization process.

Algorithm 1 Drone behavior in DIR

```
1: for all w ∈ W do                                          ▷ W = Waypoints set
2:     sendBeacon(id_b, t_d), ∀d = 0...5                     ▷ id_b = beacon ID, t_d = drone's timestamp
3:     P_resp ← receiveAckFromSensors()                      ▷ P_resp = set of sensors that reply to the drone
4:     for all id ∈ P_resp do
5:         grdist ← computeGroundDistance(id_s, t_d, t_s)    ▷ id_s = sensor ID, t_s = sensor's timestamp
6:         if grdist ≥ d_min then
7:             sendGroundDistance(id_s, grdist)
8:         end if
9:     end for
10: end for
```

During the flight, for each waypoint, the drone sends the message beacon along the six different orientations of beamwidth: up^+, up^-, dw^+, dw^-, hor^+, and hor^- (Line 2, Alg. 1). Specifically, when a sensor receives the drone's message from orientation ψ, for $\psi = d \cdot \pi/3$ and $d = 0, \ldots, 5$ (Line 3, Alg. 2), it first checks whether the d-th location of its register R (see Tab. 2) is empty or not. In the first case, the sensor is receiving for the first time the beacon from the orientation d. In the second case, since the sensor has already heard that beacon from orientation d, the message is ignored. When the sensor receives one orientation for the first time, it replies to the drone with an ack message (Line 6, Alg. 2). The drone infers the distance from the time of the round-trip message (Line 3, Alg. 1), and sends to the sensor the measure (Line 7, Alg. 1). The sensor stores the measure in the local register R (Line 7, Alg. 2). The trilateration is performed by the sensor when it has received three measurements (Line 11, Alg. 2).

4.4. Formal Properties of the DIR algorithm

In the following we formally analyze the properties of the DIR algorithm. First, we prove that it is possible to precisely measure any point in Q. Then, we study how to set the parameters to achieve the a-priori required localization precision.

Algorithm 2 Sensor behavior in DIR

```
1: ranged ← 0
2: while ranged < 3 do
3:     b ← waitBeacon()                          ▷ waits for the beacon
4:     d ← b.orientation
5:     if R[d] = 0 then                          ▷ checks the beacon orientation/polarity
6:         dist ← sendAck(t_s)                   ▷ receives the ground distance
7:         R[d] ← dist
8:         ranged ← ranged + 1
9:     end if
10: end while
11: P ← trilateration(R)                         ▷ performs the trilateration procedure and estimates its position P
```

Theorem 1. *Given a half beamwidth*

$$\theta = \arctan\left(\frac{I_w/2}{d_{min}}\right) \tag{7}$$

each point of Q can be measured by the three different orientations hor, up, and dw verifying the d_{min} and d_{max} constraints.

Proof. Let first consider the antenna's orientation hor^+ with $\theta = \arctan\left(\frac{I_w/2}{d_{\min}}\right)$. As observed from Fig. 11(a), each sector with orientation hor^+ is able to cover at least a rectangular of height I_w from distance $d_{\min}$ up to distance $d_{\max}$ (see Fig. 11(a)) if θ satisfies Eq. (7). Since the first scan of our path is at distance $F_x = d_{\min}/2$ from Q, it is easy to prove that any point in Q can be covered (thus, measured) satisfying the $d_{\min}$ and $d_{\max}$ constraints of Def. 1 using antenna's orientation hor^+. The same holds for the orientation hor^- because the last scan is outside Q, and precisely $F_x = d_{\min}/2$ beyond Q.

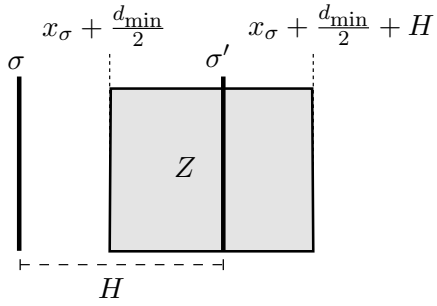


Figure 10: The stripe Z of Q parallel to scan S

For the other orientation of the antennas, consider a scan S and the antennas at its waypoints. We aim to prove that the orientations dw and up are able to cover the stripe Z (see Fig. 10) of Q parallel to σ that starts at distance $d_{\min}/2$ and finishes at distance $d_{\min}/2 + H$ from σ, without leaving any uncovered space.

Consider now the antenna's orientation dw^+. To prove that they cover Z, we will show that the sector centered at the waypoint w_1 (see Fig. 11(b)) intercepts the segment T_1P_1 of length $\overline{T_1P_1} \geq I_w$ on the line parallel to σ and at distance $d_{\min}/2$ from σ. The same sector intersects the segment T_2P_2 of length $\overline{T_2P_2} \geq I_w$ on the line parallel to σ and at distance $d_{\min}/2 + H$ from σ.

Let us start with $\overline{T_1P_1}$. By applying simple trigonometric rules to the triangles $w_1Q_1T_1$ and $w_1P_1T_1$, where $Q_1 = (d_{\min}/2, y_{w_1})$, we obtain

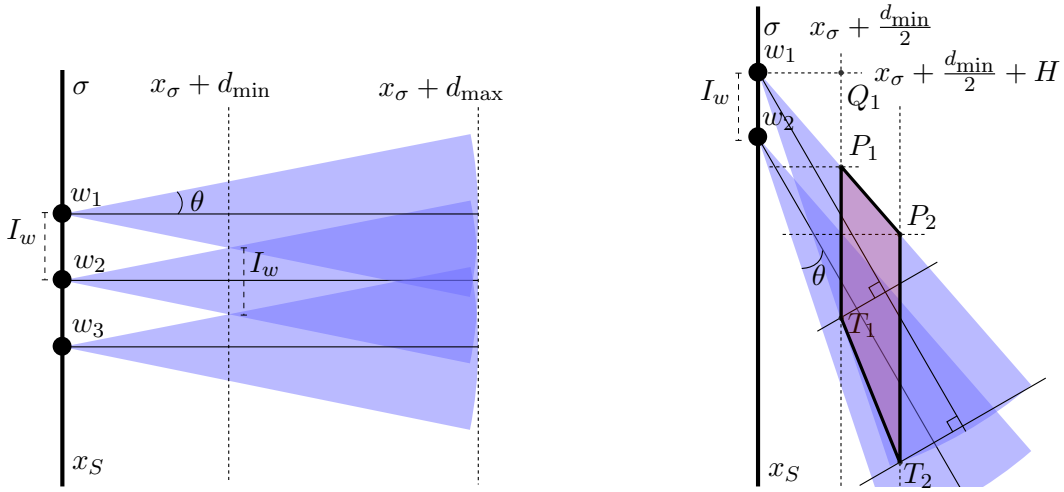
$$\overline{T_1P_1} = \overline{Q_1T_1} - \overline{Q_1P_1}$$

where

$$\overline{Q_1T_1} = \frac{d_{\min}}{2} \cot\left(\frac{\pi}{6} - \theta\right),$$
$$\overline{Q_1P_1} = \frac{d_{\min}}{2} \tan\left(\frac{\pi}{3} - \theta\right).$$

After algebraic manipulations we find $\overline{T_1P_1} > I_w$ when θ satisfies Eq. (7). Then comparing the two similar triangles $w_1T_1P_1$ and $w_1T_2P_2$, we learn that $\overline{T_2P_2} > \overline{T_1P_1} > I_w$. Hence, repeating the same reasoning for all the waypoints of σ, we conclude that the sectors with orientation dw^+ centered at the waypoints on the static path σ cover Z.

Not only all the sensors that fall in Z are covered, but also they can be measured from the antennas with orientation dw^+ satisfying the $d_{\min}$ and $d_{\max}$ constraints. Indeed, the slant distance from w_1 to any point $P = (x_P, y_P)$ that falls in Z when projected on the ground is equal to the double of the distance $x_P - x_\sigma$ from σ to P. Thus, since $d_{\min} \leq 2(x_P - x_\sigma) \leq d_{\max} - 2I_w$, the $d_{\min}$ and $d_{\max}$ constraints are verified.



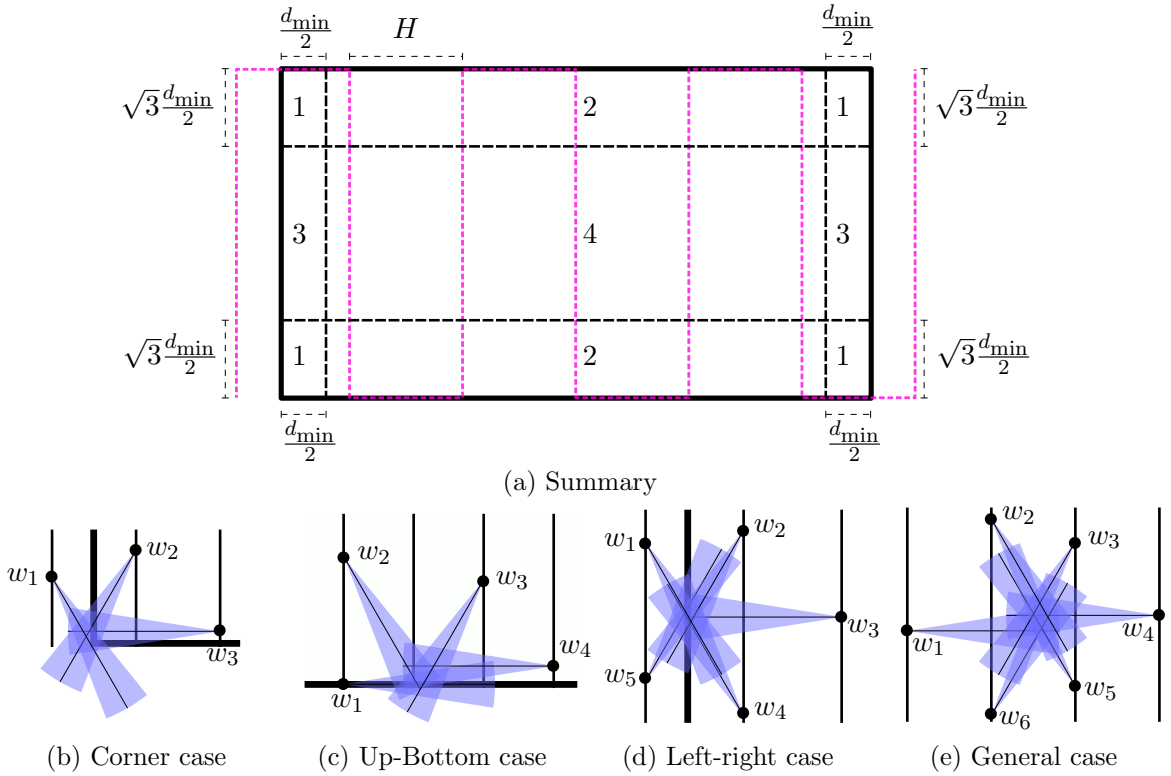
(a) Horizontal pattern (b) Down pattern

Figure 11: The antenna beam

A similar reasoning can be repeated for the other remaining orientations.

The previous discussion has proved that the largest part of Q can be reached by all the six different orientations. However, on the border of Q some orientations can be missing. We divide the analysis in four different cases, as illustrated in Fig. 12(a).

Case 1. A sensor $P = (x_P, y_P)$ placed near the bottom-left corner of Q, with $x_P < d_{\min}/2$ and $y_P < \sqrt{3}d_{\min}/2$, can be ranged only by the three waypoints, as depicted in Fig. 12(b), i.e., dw^+, hor^- and up^-. Since $F_x = d_{\min}/2$, sensor P cannot not be ranged from a horizontal beam hor^+ along the first scan because the $d_{\min}$ constraint is not satisfied. Hence the sensor has to wait until the drone passes through w_3 for collecting the *hor* type of orientation. A similar reasoning holds for the other areas labeled 1 in Fig. 12(a).



(a) Summary

(b) Corner case (c) Up-Bottom case (d) Left-right case (e) General case

Figure 12: Ranging cases

Case 2. A sensor $P = (x_P, y_P)$ placed near the bottom border of the area (except corners), with $x_P \geq d_{\min}/2$ and $y_P < \sqrt{3}d_{\min}/2$, can be ranged by four waypoints, as shown in Fig. 12(c), i.e., dw^+, $hor^{+/-}$ and up^-. For the drone it is impossible to reach P from up^+ or dw^+. Nevertheless, as can be seem in Fig. 12(c), the sensor can be always reached by the three different types of orientations and thus it can collect sufficient measurements for localizing itself. A similar reasoning can be repeated for the other areas labeled 2 in Fig. 12(a).

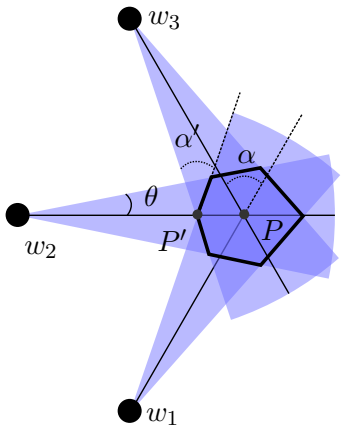
Case 3. A sensor P placed in the area of Q labeled 3 near the left border of the area (except corners), with $dw^{+/-}$, hor^- and $up^{+/-}$, can be ranged up to five waypoints, as shown in Fig. 12(d). The sensor will be localized as soon as it has collected three different types of orientations. Like the first case, to have a measure at distance $d_{\min}$ in the *hor* type of orientation, the sensor has to wait until the drone passes through w_3.

Case 4. Finally, a sensor P placed in the middle of Q, in the zone labeled 4, could be ranged by six waypoints, as shown in Fig. 12(e), i.e., $dw^{+/-}$, $hor^{+/-}$ and $up^{+/-}$. Since after measurements the three different types of orientations have been collected, the trilateration procedure starts, after three measurements in the best case, and, after five measurements, in the worst case.

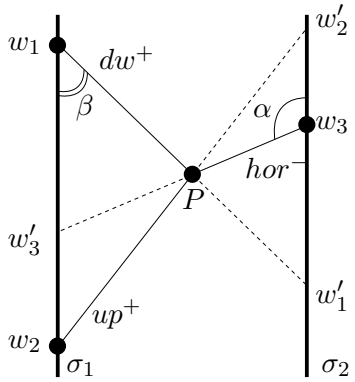
From the analysis of the Cases 1, 2, 3 and 4 and from the previous discussion on stripe Z, the correctness of Th. 1 follows. □

It is worthy to note that, although in Th. 1 the beamwidth θ is given, it is a function of the algorithm parameters that determine the localization precision. So, to ensure different levels of precision, the algorithm requires the ability of adjusting the antenna beamwidth.

As explained in Sec. 3, the localization error strongly depends on the position of the three waypoints from which the point is ranged. From Eq. (6), the error in the position of the sensor is minimum when for each sensor $\beta_{\min} = \pi/3$.



(a) The polygon measured by up^+, hor^+, and dw^+



(b) Relation between $\beta_{\min}$ and θ

Figure 13: The measured angles.

In our localization technique, the minimum angle at P is $\pi/3$ if and only if the sensor P resides at the intersection of the orientations of the three sectors centered at the waypoints $w_1(P)$, $w_2(P)$ and $w_3(P)$ (see Fig. 13(a)). Although it is not possible to achieve for every point P that the minimum angle is $\pi/3$, we can claim a strong relationship between $\beta_{\min}$ and θ.

Theorem 2. *Given the half beamwidth θ according to Th. 1, the algorithm* Dir *provides*

$$\beta_{min} \geq \frac{\pi}{3} - 2\theta \tag{8}$$

Proof. Consider a point P along with its three ranging waypoints w_1, w_2 and w_3 which belongs to two scans, say σ_1 and σ_2. Without loss of generality, let w_1 be the center of the sector dw^+, w_2 the center of the sector up^+ and w_3 the center of hor^- (see Fig. 13(b)).

Let w_i' for $i = 1, 2, 3$ be at the intersection of the extensions of the line w_iP and of the opposite scan. The turn angle at P is divided in six angles by the lines w_iP and their extensions. The width of the angles at P depends on the angles formed by the lines w_iP or their extensions $w_i'P$ and the scans. Since the line w_iP, for $i = 1, 2, 3$, belongs to the sector of w_i, the angles formed at the scans yield:

$$\frac{\pi}{2} - \theta \leq \alpha \leq \frac{\pi}{2} + \theta$$

if w_i is the center of a sector of type hor, and

$$\frac{\pi}{6} - \theta \leq \beta \leq \frac{\pi}{6} + \theta$$

if w_i is the center of a sector of type up/dw. By simple geometric considerations, the minimum angle at P is no smaller than:

$$\left\{\pi - \frac{\pi}{2} - \frac{\pi}{6} - 2\theta, 2\frac{\pi}{6} - 2\theta\right\} = \frac{\pi}{3} - 2\theta$$

Hence, $\beta_{\min} \geq \frac{\pi}{3} - 2\theta$. □

Corollary 1. *All the points in Q can be ϵ_L-precisely localized.*

Proof. It follows directly from Def. 1, and by Thms. 1 and 2. □

Finally observe that, increasing $d_{\min}$, a higher precision ϵ_L is achieved. From Eq. (7), when $d_{\min}$ increases, θ decreases. From Th. 2, $\beta_{\min}$ increases, and according to Eq. (6) the localization error becomes smaller.

Fig. 14 depicts the heatmap of minimum angle referred to the example shown in Fig. 13(a). The polygon represents the intersection of the three circular sectors, that is the points of Q that are measured by w_1, w_2 and w_3. For each point P, a "hot" color means a good minimum angle at P while a "cold" color means a bad angle. The worst point is the leftmost point P' of polygon whose $\beta_{\min} = \alpha$ in Fig. 13(a). Note that the lines w_3P' and w_1P' form with the scan that contains w_1 and w_3 angles of width $\pi/6 - \theta$, and thus $\alpha = 2\pi/6 - 2\theta$. In the example, $\theta = \arctan \frac{I_w/2}{d_{\min}} = 10$ degrees and the heatmap registers angles greater than $60 - 20 = 40$ degrees, according to Th. 2.

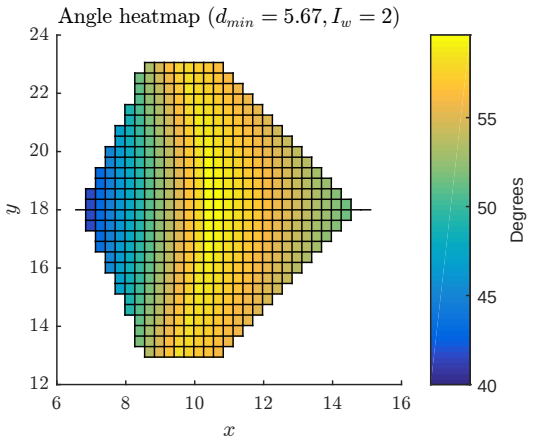


Figure 14: Minimum angle expressed (in degrees) in Dir

We are now in a position to rewrite Eq. (6) as a function that depends on h, $d_{\min}$ and I_w. Inverting this expression we can ensure the precision required by the final user.

Theorem 3. *The localization precision can be expressed as:*

$$\epsilon_L = \epsilon_s \cdot \sqrt{1 + \frac{h^2}{{d_{min}}^2}} \cdot \frac{2\sqrt{1 + \left(\frac{I_w/2}{d_{min}}\right)^2}}{1 - \sqrt{3}\left(\frac{I_w/2}{d_{min}}\right)} \tag{9}$$

Proof. By Th. 2, $\frac{\beta_{\min}}{2} \geq \frac{\pi}{6} - \theta$. The largest error occurs when the angle $\frac{\beta_{\min}}{2}$ assumes the minimum value, i.e., $\frac{\pi}{6} - \theta$. Recalling that $\theta = \arctan\left(\frac{I_w/2}{d_{\min}}\right)$, calculating $\sin\left(\frac{\pi}{6} - \theta\right)$ and substituting in Eq. (6), it holds:

$$\epsilon_L = \epsilon_s \cdot \frac{\sqrt{1 + \frac{h^2}{{d_{\min}}^2}}}{\sin\left(\frac{\pi}{6} - \theta\right)} = \epsilon_s \cdot \sqrt{1 + \frac{h^2}{{d_{\min}}^2}} \cdot \frac{2\sqrt{1 + \left(\frac{I_w/2}{d_{\min}}\right)^2}}{1 - \sqrt{3}\left(\frac{I_w/2}{d_{\min}}\right)}$$ □

In the next section, we present the OMNI algorithm which requires a more complex strategy given that only omnidirectional antennas are available. In fact, to select a waypoint in a certain direction, in OMNI, we will invoke a trilateration procedure with the first three measurements collected for each sensor in conjunction with a tessellation technique of the entire deployment area. So the interest in OMNI is twofold: OMNI can be considered an independent algorithm for localization, or a way to emulate the DIR algorithm with a minimum hardware requirement.

Finally, the fact that both the algorithms OMNI and PRECISEVERIFIERBEE [12], which are both based on omnidirectional antennas, require two trilaterations instead of a single one to achieve precise localization stresses the fact that the directional and the omnidirectional antennas act differently when the main objective is precision.

5. The OMNI algorithm

In this section we present OMNI, which does not require specialized hardware such as directional antennas. Conversely, OMNI is based on a standard omnidirectional antenna, commonly available for off-the-shelf drones [25].

The omnidirectional antenna send beacons with the drone's position isotropically. Since the sensors cannot distinguish from which direction they receive the beacon, it is difficult to select the waypoints that guarantee the desired geometry. For this reason, OMNI adopts a two phase approach. In the first phase, the sensors obtain a rough estimation of their position. In the second phase they use this estimation to pick among all the available waypoints the best ones, called *precise waypoints*, that equally divide the turn angle around the sensor itself to perform a trilateration and obtain a precise localization.

The rest of the section is structured as follows. In Sec. 5.1 we introduce the static path calculated for OMNI, and in Sec. 5.2 we describe the localization process.

5.1. Static Path

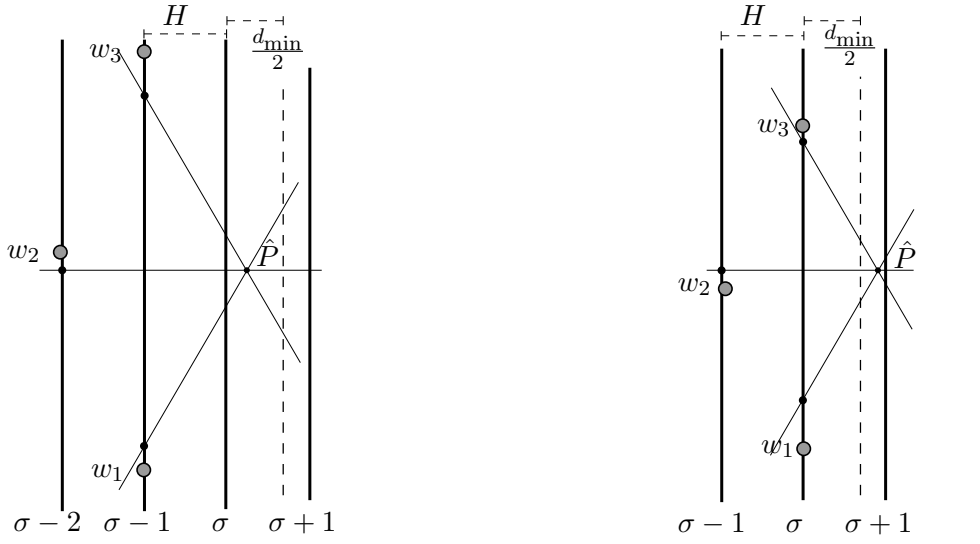
The static path Π_O is depicted as a dashed line in Fig. 3. According to the description given in Sec. 3.1, the drone's mission starts at $F = (-F_x, -F_y)$, where $F_x = \frac{d_{\min}}{2} + H$, $F_y = (d_{\max} - I_w)\frac{\sqrt{3}}{2}$. The inter-scan distance is set to $H = (d_{\max} - d_{\min} - 2I_w)\frac{1}{2}$. The path Π_O starts with two vertical scans outside Q that are used to measure the sensors in the leftmost stripe of Q close to the vertical border, while the last vertical scan is at distance no larger than $\left\lfloor \frac{d_{\max}-2I_w}{2} \right\rfloor$ from the rightmost border. Accordingly, Π_O finishes at $E = (E_x, E_y)$, where $E_x \leq Q_x + \left\lfloor \frac{d_{\max}-2I_w}{2} \right\rfloor$ and y_E can be either $-F_y$ or $Q_y + F_y$ depending on the parity of the number of vertical scans.

5.2. Localization Process

The behavior of the drone under OMNI is the same as during algorithm DIR (see Algorithm 1). However, the behavior of the sensors is quite different, and it is summarized in Algorithm 3.

According to OMNI, until the first trilateration, the sensor stores all the measurements that it receives which are above the $d_{\min}$ ground distance. As soon as the sensor has collected three distance measurements which belong to two different scans and are greater than the minimum ground distance, the sensor performs the first trilateration (Line 10, Alg. 3) to compute a rough estimation of its position $\hat{P} = (x_{\hat{P}}, y_{\hat{P}})$. Note that the measurements used for the first trilateration already partially satisfy the requirements in Def. 1. However, we still do not have any guaranteed bound on the position error because the width of $\beta_{\min}$ is not known.

Then, from $\hat{P}$, the sensor locates the closest vertical scan σ on its left. Then, conceptually, the sensor derives the six rays that equally divide the turn angle around its current position. The intersections of such rays with the scan σ on the left of $\hat{P}$ are then the most desirable positions as regard to the waypoint geometry (i.e., the $\beta_{\min}$ constraint) from where taking the final measurements. However, such points may not coincide with waypoints because the drone takes measurements at regular distance and may also not satisfy the $d_{\min}$ constraint (see Fig. 15). So, to find the three precise waypoints obeying all the constraints, the sensor locally computes the three precise waypoints $w_1(\hat{P}), w_2(\hat{P}), w_3(\hat{P})$ on the scans on its left according to Eq. (11) and Eq. (12) (Line 11, Alg. 3). Selected $w_1(\hat{P}), w_2(\hat{P}), w_3(\hat{P})$, the sensor continues to hear to the drone until it has collected all the three precise waypoints and discards the measurements which are useless. Finally, the sensor computes the second precise trilateration using the precise waypoints (Line 14, Alg. 3), and the localization process is finished.



(a) $x_\sigma \leq x_{\hat{P}} \leq x_\sigma + \frac{1}{2}d_{\min}$ (b) $x_\sigma + \frac{1}{2}d_{\min} < x_{\hat{P}} \leq x_{\sigma+1}$

Figure 15: Rays that divide the turn angle in $\hat{P}$ and the associated precise waypoints.

In the following we detail as the precise waypoints $w_1(\hat{P}), w_2(\hat{P}), w_3(\hat{P})$ are defined. Let us denote the ordinates of the intersection of the rays that equally divide the turn angle in $\hat{P}$ and the vertical scan σ as:

$$\begin{aligned} \bar{y}_1(\sigma) &= \left\lfloor \frac{y_{\hat{P}} - \sqrt{3}(x_{\hat{P}} - x_\sigma)}{2I_w} \right\rfloor 2I_w \\ \bar{y}_3(\sigma) &= \left\lceil \frac{y_{\hat{P}} + \sqrt{3}(x_{\hat{P}} - x_\sigma)}{2I_w} \right\rceil 2I_w \\ \bar{y}_2(\sigma) &= \frac{\bar{y}_1(\sigma) + \bar{y}_3(\sigma)}{2} \end{aligned} \tag{10}$$

Note that the floor operation is to approximate the ordinates of the intersections so as they fall at waypoints[1].

[1]The use of $2I_w$ instead of I_w in Eq. (10) is to force also the middle ordinate $\bar{y}_2(\sigma)$ to fall at a waypoint.

Finally, depending on the distance between $\hat{P}$ and the leftmost scan, the precise waypoints are so defined (see Fig. 15):
If the sensor $\hat{P}$ is very close to the closest vertical scan on its left σ, i.e., if $x_\sigma \leq x_{\hat{P}} \leq x_\sigma + \frac{1}{2}d_{\min}$, the sensor must select the ranging waypoints on the vertical scans $\sigma - 1$ and $\sigma - 2$.
The three precise waypoints are then:

$$\begin{aligned} w_1(\hat{P}) &= (x_{\sigma-1}, \bar{y}_1(\sigma-1)) \\ w_2(\hat{P}) &= (x_{\sigma-2}, \bar{y}_2(\sigma-2)) \\ w_3(\hat{P}) &= (x_{\sigma-1}, \bar{y}_3(\sigma-1)) \end{aligned} \tag{11}$$

If $x_\sigma + \frac{1}{2}d_{\min} < x_{\hat{P}} \leq x_{\sigma+1}$, the sensor can select the ranging waypoints directly on σ and $\sigma - 1$.
In this case, the three precise waypoints are:

$$\begin{aligned} w_1(\hat{P}) &= (x_\sigma, \bar{y}_1(\sigma)) \\ w_2(\hat{P}) &= (x_{\sigma-1}, \bar{y}_2(\sigma-1)) \\ w_3(\hat{P}) &= (x_\sigma, \bar{y}_3(\sigma)) \end{aligned} \tag{12}$$

In Sec. 5.3, we will prove that the precise waypoints satisfy Def. 1.

Algorithm 3 Sensor behavior in OMNI

```
1: ranged ← 0
2: D[...] ← ∅                                   ▷ D: the set of measurements for the sensor
3: while P = ∅ do
4:     b ← waitBeacon()
5:     W ← b.waypoint
6:     dist ← sendAck(t_s)
7:     D[W.id] ← dist
8:     ranged ← ranged + 1
9:     if ranged = 3 then
10:        P̂ ← trilateration(D)                  ▷ rough estimation of P
11:        w1(P̂), w2(P̂), w3(P̂) ← computeOptimalTriple(()P̂)   ▷ using Eq. (11) or Eq. (12)
12:    end if
13:    if w1(P̂), w2(P̂), w3(P̂) are in D then
14:        P ← trilateration(w1(P̂), w2(P̂), w3(P̂))   ▷ second estimation of P
15:    end if
16: end while
```

5.3. Formal Properties of the OMNI algorithm

Similarly to the DIR algorithm, in this section we prove that OMNI is able to correctly localize any point in Q and that the algorithm is able to provide an a-priori required localization precision. Let the vertical scans in Π_O be numbered from 0 to ν. The construction of the tessellation that is used to prove the correctness of OMNI algorithm is illustrated in Fig. 16. Consider in Π_O two consecutive vertical scans σ^{i-1} and σ^i, with $1 \leq i \leq \nu$. Let $x_{\sigma^i} = x_F + (i-1)H$ be the x-coordinate of the vertical scan σ^i of Π_O. Given i, let M_R^i be the stripe of Q whose x-coordinates belong to the interval $\left[x_{\sigma^i} + \frac{d_{\min}}{2}, x_{\sigma^i} + \frac{d_{\max}-2I_w}{2}\right]$, as illustrated in Fig. 16.

Now we will concentrate on proving that the localization of the points in M_R^i is correct. During the proof we will work on a set of discrete points of the plane: the selection of the precise waypoints

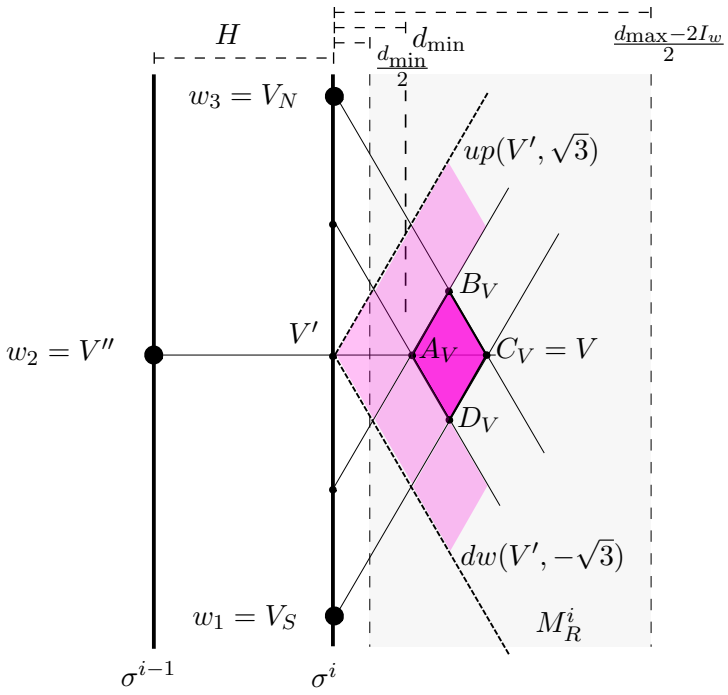


Figure 16: The tessellation and, in gray, the stripe M_R^i

is done for such discrete sensors of M_R^i and then extended to the remaining points. For this aim, we introduce a tessellation of M_R^i so defined.

Consider the *even* waypoints, that is the waypoints at distance $2I_w$ on the vertical scans. From every even waypoint $w = (x_w, y_w)$ on σ^i, we draw the two lines that pass through w with slope $m_1 = \sqrt{3}$ and $m_2 = -\sqrt{3}$ denoted, respectively, as $up(w, m_1)$ and $dw(w, m_2)$. Such lines design a lattice on M_R^i, made of diamond shapes, denoted as $\diamond$. The intersection point of the lines *up* and *dw* in M_R^i gives the vertex V. V is a special point in M_R^i because the intersections V_S and V_N of the two lines *up* and *dw* with the vertical scan σ^i are two of the three precise waypoints for V. The third precise waypoint of V, denoted by V'', is the projection of V on σ^{i-1}. Each vertex $V = (x_V, y_V)$ of the lattice is associated with the diamond $\diamond(V)$ on its left. The vertices of $\diamond(V)$ are: $C_V = V$, $D_V = (x_V - \frac{1}{\sqrt{3}}I_w, y_V - I_w)$, $A_V = (x_V - \frac{2}{\sqrt{3}}I_w, y_V)$, and $B_V = (x_V - \frac{1}{\sqrt{3}}I_w, y_V + I_w)$. All the diamonds have the same size: $\overline{A_V C_V} = \frac{2}{\sqrt{3}}I_w$ and $\overline{B_V D_V} = 2I_w$.Note that since $\overline{B_V D_V} = 2I_w$, the projection V'' of V coincide with a waypoint.

The main reason we introduced the tessellation is that the waypoints $w_1 = V_S$, $w_2 = V''$, and $w_3 = V_N$ used for localize V are also used for any point P of the diamond $\diamond(V)$ which V is associated to, as illustrated in Fig. 17. For notation easiness, we simply denote the waypoints $w_1(V)$, $w_2(V)$, and $w_3(V)$ as w_1, w_2, and w_3 in the following discussion.

Theorem 4. *Given a vertex V of the logical tessellation of M_R^i, each point $P \in \diamond(V)$ can be measured by verifying the d_{min} and d_{max} constraints.*

Proof. The sensors $P \in \diamond(V)$ are ranged by the three waypoints $w_1(P) = V_S = w_1$, $w_2(P) = V'' = w_2$, and $w_3(P) = V_N = w_3$. To prove the $d_{\min}$ and $d_{\max}$ constraints for each internal sensor $P \in \diamond(V)$, first we consider the trivial case $P = V$, then we consider each sensor $P \in \diamond(V)$.

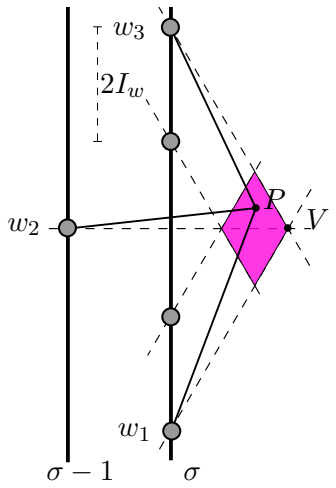


Figure 17: A point P inside the diamond $\diamond(V)$ and its ranging waypoints.

Without loss of generality, we take into account only the analysis of w_1, since is symmetric to w_3.

Case $P = V$. Considering the vertex V, by geometric arguments it holds that $\overline{V_S V} = \overline{V_N V} = 2\overline{V'V} = 2(x_V - x_{\sigma^i})$, where V' is the projection of V on σ^i. Thus, recalling that V belongs to the stripe M_R^i on the right of σ^i, that is, $\frac{d_{\min}}{2} \leq x_V - x_{\sigma^i} \leq \frac{d_{\max} - 2I_w}{2}$, it follows that:

$$d_{\min} \leq \overline{w_1 V} \leq d_{\max} - 2I_w$$

and

$$d_{\min} < H + \frac{d_{\min}}{2} < \overline{w_2 V} \leq H + \frac{d_{\max} - 2I_w}{2} < d_{\max}.$$

Thus, the $d_{\min}$ and $d_{\max}$ constraints are satisfied for w_1 and w_2, and w_3 for V.

Case $P \in \diamond(V)$. Let us first consider each point P in the lower half-diamond called $\nabla(V) = A_V C_V D_V$. Let P' be the projection of P on σ^i and let $\measuredangle P' V_S P$ be the angle at V_S between the ray $V_S P$ and the vertical scan σ^i. Since $\overline{V_S P'} \leq \overline{V_S V'}$ because P belongs to the triangle $A_V D_V C_V$ and $\measuredangle P' V_S P \leq \measuredangle V' V_S V$, it holds that:

$$\overline{w_1 P} = \frac{\overline{w_1 P'}}{\cos(\measuredangle P' w_1 P)} \leq \frac{\overline{w_1 V'}}{\cos(\measuredangle P' w_1 P)} \leq \frac{\overline{w_1 V'}}{\cos(\measuredangle V' w_1 V)} = \overline{w_1 V} \leq d_{\max} - 2I_w$$

Moreover, for each point P in the upper half-diamond $\Delta(V) = A_V B_V C_V$, consider the point $P'' \in \nabla(V)$ symmetric to P with respect to the line $A_V C_V$. By the triangle inequality, given that V belongs to M_R^i and $\overline{P'' P} \leq \overline{B_V D_V}$, we have:

$$\overline{w_1 P} \leq \overline{w_1 P''} + \overline{P'' P} \leq \overline{w_1 P''} + 2I_w \leq \overline{w_1 V} + 2I_w \leq d_{\max}$$

Thus, the $d_{\min}$ and $d_{\max}$ constraints are satisfied for w_1 and w_3 for $P \in \diamond(V)$.

With regard to w_2, let Z be the projection of $P \in \diamond(V)$ on the diamond diagonal $A_V C_V$. Moreover, since $H > \frac{d_{\min}}{2}$, it easily follows that:

$$\overline{w_2 P} \leq \overline{w_2 Z} + \overline{Z P} \leq \overline{w_2 V} + I_w < H + \frac{d_{\max} - 2I_w}{2} < d_{\max}$$

and

$$\overline{w_2P} \geq \overline{w_2Z} \geq H + \frac{d_{\min}}{2} > d_{\min}.$$

Thus, the $d_{\min}$ and $d_{\max}$ constraints are satisfied for w_2 for $P \in \diamond(V)$. □

As explained in Sec. 3, the localization error strongly depends on the position of the three precise waypoints w_1 and w_2, and w_3 from which the point P is ranged.

Theorem 5. *Given the vertex V,* Omni *provides a value for β_{min} such that*

$$\beta_{min} \gtrapprox 2 \arctan\left(\frac{\frac{d_{min}}{2}}{\sqrt{3}\frac{d_{min}}{2} + 2I_w}\right) \tag{13}$$

Proof. For the vertex V, as said, the tessellation divides the turn angle 2π in V in six equal angles. Namely, the intersections of the rays that start from V by construction coincide with waypoints and satisfy:

$$\tan \beta(V) = \frac{x_V - x_\sigma}{y_{w_3} - y_V} = \sqrt{3}.$$

Thus, $\beta_{\min}(V) = \frac{\pi}{3}$. For each other point $P \in \Delta(V)$, the turn angle is asymmetrically divided. The largest error occurs in the proximity of A_V. Namely the points in the proximity of A_V,

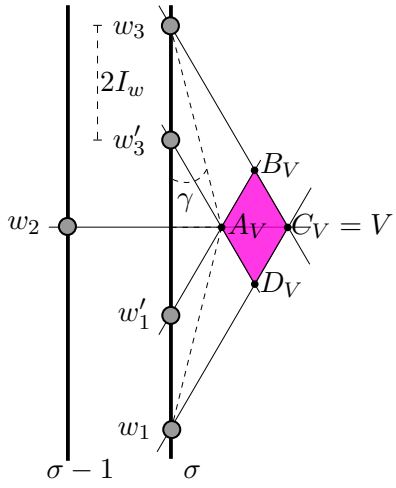


Figure 18: The worst case for $\beta_{\min}$

and in particular A_V, would be almost equally divided if they were measured starting from the two adjacent waypoints $w'_3(V)$ and $w'_1(V)$, at distance $2I_w$, from $w_3(V)$ and $w_1(V)$ (see Fig. 18). However, since they are measured by $w_3(V)$ and $w_1(V)$ because they are in $\diamond(V)$, their turn angle is unsymmetrically divided. The angle asymmetry in the points in the proximity of A_V when measured by $w_3(V)$, $w_2(V)$, and $w_1(V)$ is bounded by the asymmetry occurring at A_V when measured by $w_3(V)$, $w_2(V)$, and $w_1(V)$. The distance of the selected waypoint with respect to the optimal waypoint for A_V is $2I_w$. The impact of such an $2I_w$ difference on the selected waypoints leads to a localization error which is greater when the diamond is close to the vertical scan because $y_{w_1} - y_{A_V}$ or $y_{w_3} - y_{A_V}$ is smaller if A_V is close to the vertical scan. Hence the smallest angle

$\beta(A_V)$ in A_V when the turn angle in A_V is divided with the rays that connect A_V with $w_3(V)$, $w_2(V)$ and $w_1(V)$ has width no smaller than $\beta(A_V) \geq \pi - 2(\frac{\pi}{2} - \gamma) = 2\gamma$, where

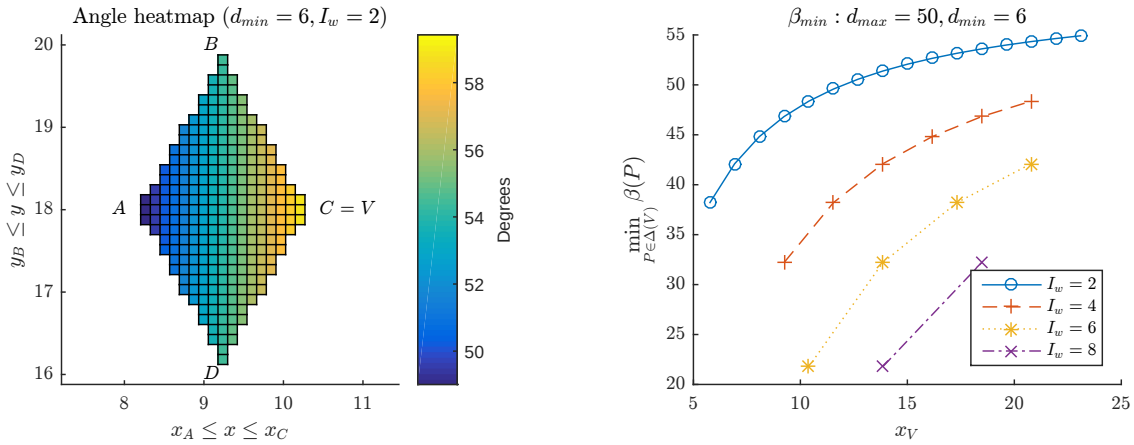
$$\arctan(\gamma) \approx \frac{\frac{d_{\min}}{2}}{\sqrt{3}\frac{d_{\min}}{2} + 2I_w}.$$

In conclusion, for each $P \in M_R^i$, $\beta_{\min}$ occurs in the proximity of the vertices A_V with minimum x-coordinate in M_R^i. Therefore,

$$\frac{\beta_{\min}}{2} \gtrapprox \arctan\left(\frac{\frac{d_{\min}}{2}}{\sqrt{3}\frac{d_{\min}}{2} + 2I_w}\right) \tag{14}$$

□

Fig. 19(a) illustrates the width of the angle $\beta(P)$ when P moves in a diamond shape of M_R^i and gives evidence that $\beta(A_V) = \min_{P \in \Diamond(V)} \beta(P)$. Fig. 19(b) shows the fact that increasing the distance of A_V from σ^i, $\beta(A_V)$ increases. As claimed, the diamond at distance $\frac{d_{\min}}{2}$ to the vertical scan is the one that gives the worst bound for $\beta_{\min}$.



(a) Minimum angle expressed (in degrees) in Omni

(b) $\beta(A_V) = \min_{P \in \Diamond(V)} \beta(P)$ versus the x_V

Figure 19: Minimum angle analysis in Omni

Using the result in Theorems 4 and 5, we have all the informations required to apply Def. 1 in each diamond in M_R^i and thus in all the stripe M_R^i, as shown by the following Corollary.

Corollary 2. *Each point of Q can be correctly measured according to Def. 1.*

Proof. Since Th. 4 has proved that from σ^{i-1} and σ^i the sensors in M_R^i are correctly measured, considering all the pairs of consecutive scans $(\sigma^0, \sigma^1), (\sigma^1, \sigma^2), \ldots, (\sigma^{\nu-1}, \sigma^\nu)$ we cover $\cup_{i=1}^{\nu} M_R^i = Q$. Since Th. 5 has proved the $\beta_{\min}$ constraint for each diamond in M_R^i, we can conclude that all the sensors in Q can be correctly measured from the vertical scans $\sigma^0, \ldots, \sigma^\nu$. □

Now, from Eq. (6) in Def. 1, we are ready to state the following result:

Theorem 6. *The localization precision in* Omni *can be expressed as:*

$$\epsilon_L = \epsilon_s \cdot \sqrt{1 + \frac{h^2}{{d_{min}}^2}} \cdot \frac{\sqrt{1 + \tan^2\left(\frac{\beta_{min}}{2}\right)}}{\tan\left(\frac{\beta_{min}}{2}\right)} \tag{15}$$

Proof. By Th. 5, $\frac{\beta_{\min}}{2} = \arctan\left(\frac{\frac{d_{\min}}{2}}{\sqrt{3}\frac{d_{\min}}{2}+2I_w}\right)$. The largest error occurs when the angle $\frac{\beta_{\min}}{2}$ assumes the minimum value, so substituting in Eq. (6), it holds:

$$\epsilon_L = \epsilon_s \cdot \frac{\sqrt{1 + \frac{h^2}{{d_{\min}}^2}}}{\sin\left(\frac{\beta_{\min}}{2}\right)} = \epsilon_s \cdot \sqrt{1 + \frac{h^2}{{d_{\min}}^2}} \cdot \frac{\sqrt{1 + \tan^2\left(\frac{\beta_{\min}}{2}\right)}}{\tan\left(\frac{\beta_{\min}}{2}\right)} \qquad \square$$

In the next section, we simulate the Dir and Omni algorithms and we compare their performance.

6. Experimental Evaluation

We have implemented the new Dir and Omni localization algorithms in MATLAB programming language. In the following we compare their performance under a variety of different conditions. Due to the lack of results in literature on algorithms that ensure the final user localization adopting a drone, we only compare Dir and Omni.

6.1. The Parameters to Derive the User-required Localization Precision

To achieve the user-required localization precision ϵ_L, given h and r, we can invert Eq. (9) to find pairs of values $d_{\min}$, I_w that guarantee ϵ_L. Tab. 3(a) and Tab. 3(b) give the $d_{\min}$ values, respectively, for Dir and Omni, for different input values of ϵ_L fixed $I_w = \{2, 5, 10\}$, $h = \{15, 30\}$ and $r = 150$.

Table 3: The $d_{\min}$ values when $r = 150$

(a) for Dir

		ϵ_L		
h	I_w	0.3	0.6	0.9
15	2	16.75	7.35	5.33
	5	22.43	10.62	8.31
	10	33.25	16.41	13.43
30	2	30.06	12.61	8.72
	5	35.28	15.77	11.64
	10	44.86	21.25	16.62

(b) for Omni

		ϵ_L		
h	I_w	0.3	0.6	0.9
15	2	19.29	8.22	5.83
	5	27.96	11.39	8.16
	10	43.42	15.90	11.22
30	2	32.82	13.86	9.63
	5	41.44	17.61	12.55
	10	55.93	22.78	16.32

As one can see, given the same parameters, the no-measurement area, given by the $d_{\min}$ constraint, is comparable for Omni and Dir. Only when I_w increases and the final user requires a higher precision, Omni pays a stronger $d_{\min}$ constraint than Dir. In fact, if I_w increases, the size of the diamond increases and Omni becomes inherently more imprecise than Dir. Thus, a stronger $d_{\min}$ constraint is required to balance the imprecision due to I_w.

6.2. Experimental Settings

In our simulations, we deploy at random n sensors, with $n = 50, 100, 250$ and 500, on a 500×500 m^2 map which corresponds to an average density of one sensor in every square of side 25 m. In addition, we set the communication radius $r = 150$ m and the altitude $h = 15$ m. For each value of n, we generate 35 different random deployments of sensors and averaged the results.

Given a user-required *localization precision* ϵ_L and selected I_w, we find the value $d_{\min}$ that ensures the selected precision for DIR and OMNI by inverting, respectively, Eq. (9) and Eq. (15).

Once all the parameters, $d_{\min}$, $d_{\max}$, h, I_w have been calculated, the localization process of both DIR and OMNI algorithms is defined. We simulate the measurement in DIR and OMNI as follows. Since the drone is simulated, we only know the exact ground distance between the waypoint and the sensor. We can calculate using the altitude the exact slant distance. Thus, to simulate the slant measure we proceed as follows: fixed the slant precision $\epsilon_s = 0.1$, as claimed by DecaWave IR-UWB [20], we compute the slant distance s measured by the drone by adding a random generated slant error $s_{\text{err}} \in [-\epsilon_s, \epsilon_s]$ to the exact slant distance sensor-drone. Then we compute the ground measure d as: $\sqrt{s^2 - h^2}$. Note that d at this point includes the ground error.

Finally, we define the *error bound* $\overline{\epsilon_L}$ as follows. For each random deployment M_i of sensors, with $i = 1, \ldots, 35$, we compute the *worst experimental localization precision*, that is, $\epsilon_L(M_i) = \max_{P \in Q}\{\epsilon_L(P)\}$, where the *experimental position precision* $\epsilon_L(P)$ for the point P is the absolute value of the difference between the exact position of P and the estimated position computed applying Eq. (1). Then, we define $\overline{\epsilon_L}$ as the *average* of the worst experimental localization precision over all the 35 random deployments, that is, $\overline{\epsilon_L} = \sum_{i=1}^{35} \epsilon_L(M_i)/35$.

Table 4: Summary of Notation for Experiments

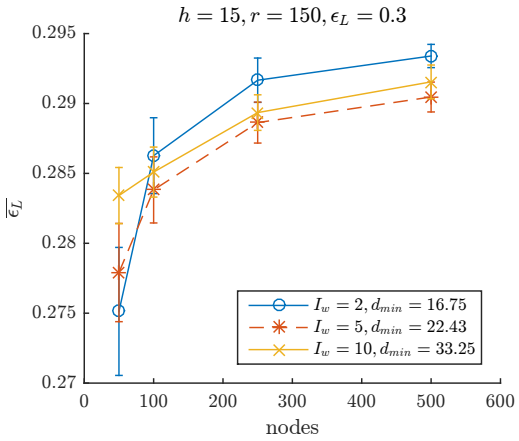
symbol	description	unit
ϵ_L	localization precision given in input as parameter	m
$\epsilon_L(P)$	experimental position precision for a point P	m
$\epsilon_L(M_i)$	worst experimental localization precision for deployment M_i	m
$\overline{\epsilon_L}$	error bound: *average* value of the worst bound over all 35 missions with the same parameters	m

6.3. Experimental Results

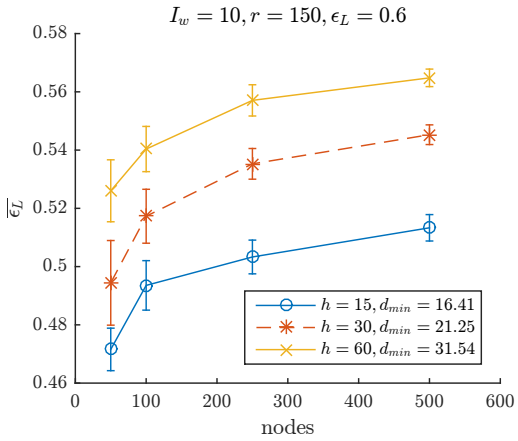
The goal of the experiments described in this section is to compare the user-defined precision ϵ_L with the experimental precision, denoted as the *error bound* $\overline{\epsilon_L}$. Specifically, we want to experimentally verify that our bound holds, and thus $\overline{\epsilon_L} < \epsilon_L$.

Given the localization precision ϵ_L, we select $d_{\min}$ for DIR by applying Eq. (9) and compare it with the results from the experiments. Fig. 20 plots the error bound $\overline{\epsilon_L}$ of the DIR algorithm for $\epsilon_L = 0.3$ and $\epsilon_L = 0.6$. In Fig. 20, the value $d_{\min}$ has been computed by inverting Eq. (9). Fig. 20(a) shows that the smallest error bound increases with I_w, except when $d_{\min}$ is very small: in that case, the ground error becomes large and dominates on the error due to the geometry. So, the minimum error for $\epsilon_L = 0.3$ is achieved for $I_w = 5$. Fig. 20(b) shows that the error bound increases with h.

It is worthy to note that the error bound in Fig. 20 is always smaller than the user-defined localization precision. Not only the average value over all the missions $\overline{\epsilon_L}$ is smaller than ϵ_L, but also the maximum error in each mission is below the user-defined localization precision, i.e., $\epsilon_L(M) \leq \epsilon_L$.

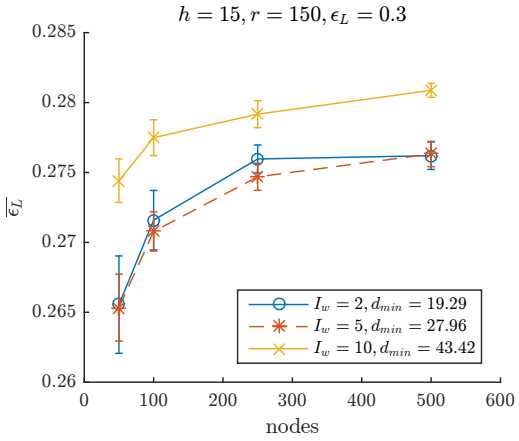


(a) $\epsilon_L = 0.3$ and I_w varies

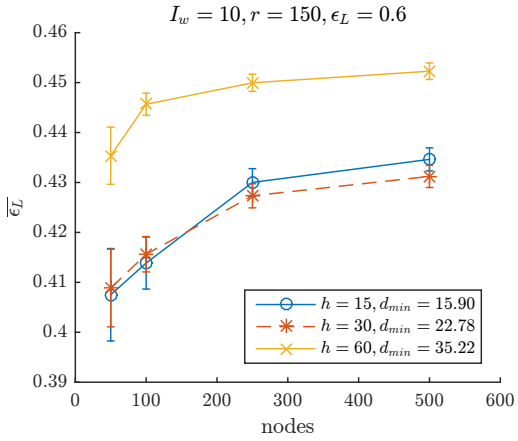


(b) $\epsilon_L = 0.6$ and h varies

Figure 20: The error bound $\overline{\epsilon_L}$ in Algorithm DIR



(a) $\epsilon_L = 0.3$ and I_w varies



(b) $\epsilon_L = 0.6$ and h varies

Figure 21: The error bound $\overline{\epsilon_L}$ in Algorithm OMNI

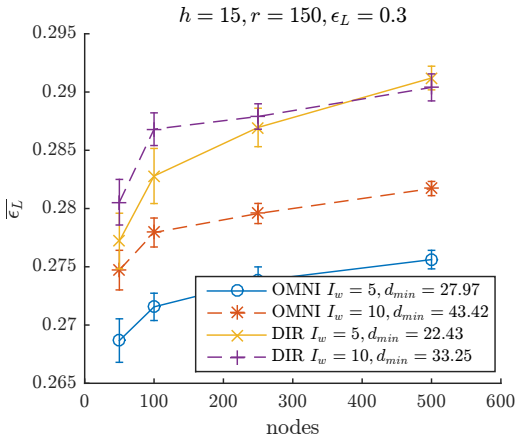


Figure 22: Comparing the error bound $\overline{\epsilon_L}$ for OMNI and DIR when $\epsilon_L = 0.3$

Fig. 21 plots the error bound $\overline{\epsilon_L}$ of the OMNI algorithm. As expected, the error bound is always smaller than the user-defined localization precision. OMNI, as DIR, is more precise when $I_w = 5$ than when $I_w = 2$. This is because the $d_{\min}$ values for $I_w = 2$ are below 20 meters for both algorithms, and so the ground error that affects the measurements is bigger in both cases when $I_w = 2$ than when $I_w = 5$.

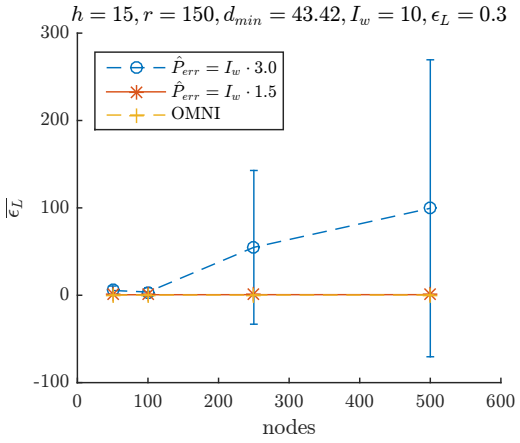


Figure 23: The error bound $\overline{\epsilon_L}$ for OMNI when the first estimation $-\hat{P}_{\text{err}} \leq \hat{P} \leq \hat{P}_{\text{err}}$

The a-priori localization precision ϵ_L is well ensured by both the proposed algorithms. Fig. 22 provides a comparison between the precision of OMNI and DIR. The OMNI algorithm is slightly more precise than DIR because OMNI uses values of $d_{\min}$ larger than DIR, and the $\beta_{\min}$ angle is larger in OMNI than DIR (see Fig. 19(a) and Fig. 14). Moreover, OMNI uses two trilaterations. However, one can note that the precision of OMNI algorithm strongly depends on the quality of the first position estimation. If the first estimation of the position is so poor that it computes a wrong diamond, the second estimation can only be poor. To validate this second motivation, in Fig. 23, we report the error bound $\overline{\epsilon_L}$ when the error of the first trilateration is accentuated (by injecting an error ad-hoc) up to $\hat{P}_{\text{err}}$. Clearly, the blows-up effect is larger when n is large because the probability of injecting a very high error increases with n. When $\hat{P}_{\text{err}} = 3I_w$, it is most likely that the first estimation returns a wrong diamond. In this case, the localization error $\overline{\epsilon_L}$ blows up significantly.

Nonetheless recall that the error in $\hat{P}$ is not arbitrary large because at least the ground error is bounded due to the fact that the $d_{\min}$ constraint is verified.

Finally, in Fig. 24, we compare the length of the static path Π_O and Π_D of the OMNI and DIR algorithms at different levels of precision, having fixed $I_w = 10$. Fixed the same level of precision, DIR and OMNI have more or less the same number of vertical scans, but each vertical scan in OMNI goes well beyond the deployment area. Precisely, in OMNI, each vertical scan has length $2F_y + Q_y$, where $F_y = (d_{\max} - I_w)\sqrt{3}/2$, whereas $F_y = 0$ for DIR. Fig. 24 shows that the static path of DIR is better than that of OMNI. The increase on the path length of the static path of OMNI is the price paid for not using specialized hardware. Note that the path length in OMNI is not monotonic decreasing because we use a fixed inter-scan value without resizing it as we do in DIR.

For each level of precision, the lost in the path length of OMNI versus DIR is above 30%. Both algorithms require a longer path if the localization error ϵ_L is smaller. When the communication radius increases, the threshold $d_{\min}$ has a smaller impact on the length path for both algorithms

because the number of vertical scans decreases.

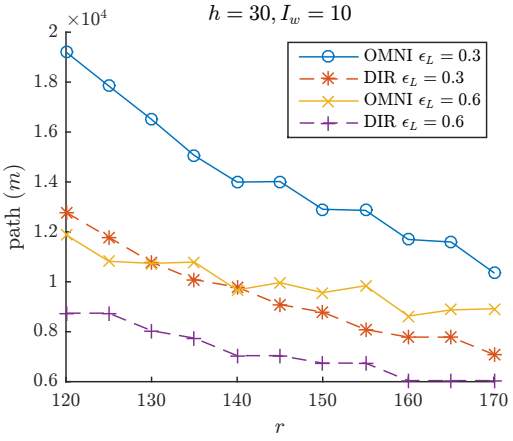


Figure 24: Path length in Omni and Dir algorithms

7. Conclusion and Future Work

In this paper, we presented two localization algorithms that replace multiple fixed anchors sensors with a flying drone. The peculiarity of our algorithms is their ability to guarantee any localization precision required by the final user of the wireless sensor network with directional and omnidirectional antennas. This result is achieved by choosing only measurements above a certain threshold $d_{\min}$, and forcing the ranging waypoints to observe a specific geometry.

Omni and Dir can be applied in any 2D environment, such as a vertical surfaces, regular slopes, etc., by taking into account the different orientation of the horizontal plane in Eq. (3). However, if the orientation is not regular, or even not known a-priori, substantial modifications to the proposed algorithms are needed. We will address these issues as future extensions of this paper.

References

[1] V. K. Shah, S. Bhattacharjee, S. Silvestri, S. K. Das, Designing sustainable smart connected communities using dynamic spectrum access via band selection, in: ACM BuildSys, Delft, the Netherlands, 2017.

[2] N. Bartolini, T. Calamoneri, T. La Porta, C. Petrioli, S. Silvestri, Sensor activation and radius adaptation (sara) in heterogeneous sensor networks, ACM Transactions on Sensor Networks (TOSN) 8 (3) (2012) 24.

[3] F. Barsi, A. A. Bertossi, F. B. Sorbelli, R. Ciotti, S. Olariu, M. C. Pinotti, Asynchronous corona training protocols in wireless sensor and actor networks, IEEE Transactions on Parallel and Distributed Systems 20 (8) (2009) 1216–1230.

[4] G. Han, H. Xu, T. Q. Duong, J. Jiang, T. Hara, Localization algorithms of wireless sensor networks: A survey, Telecommun. Syst. 52 (4) (2013) 2419–2436.

[5] N. Bartolini, S. Ciavarella, S. Silvestri, T. La Porta, On the vulnerabilities of voronoi-based approaches to mobile sensor deployment, IEEE Transactions on Mobile Computing 15 (12) (2016) 3114–3128.

[6] L. Gupta, R. Jain, G. Vaszkun, Survey of important issues in uav communication networks, IEEE Communications Surveys Tutorials 18 (2) (2016) 1123–1152.

[7] M. Flury, R. Merz, J. L. Boudec, Synchronization for impulse-radio UWB with energy-detection and multi-user interference: Algorithms and application to IEEE 802.15.4a, IEEE Trans. Signal Processing 59 (11) (2011) 5458–5472.

[8] K. Goss, R. Musmeci, S. Silvestri, Realistic models for characterizing the performance of unmanned aerial vehicles, in: Computer Communication and Networks (ICCCN), 2017 26th International Conference on, IEEE, 2017, pp. 1–9.

[9] T. He, C. Huang, B. M. Blum, J. A. Stankovic, T. Abdelzaher, Range-free localization schemes for large scale sensor networks, in: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, MobiCom '03, ACM, New York, NY, USA, 2003, pp. 81–95.

[10] C. H. Ou, K. F. Ssu, Sensor position determination with flying anchors in three-dimensional wireless sensor networks, IEEE Transactions on Mobile Computing 7 (9) (2008) 1084–1097.

[11] S. Halder, A. Ghosal, A survey on mobile anchor assisted localization techniques in wireless sensor networks, Wireless Networks 22 (7) (2016) 2317–2336.

[12] P. Perazzo, F. B. Sorbelli, M. Conti, G. Dini, C. M. Pinotti, Drone path planning for secure positioning and secure position verification, IEEE Transactions on Mobile Computing 16 (9) (2017) 2478–2493.

[13] K. Kim, W. Lee, Mbal: A mobile beacon-assisted localization scheme for wireless sensor networks, in: 2007 16th International Conference on Computer Communications and Networks, 2007, pp. 57–62.

[14] S. M. Mazinani, F. Farnia, Localization in wireless sensor network using a mobile anchor in obstacle environment, International Journal of Computer and Communication Engineering 2 (4) (2013) 438.

[15] J. Jiang, G. Han, H. Xu, L. Shu, M. Guizani, LMAT: Localization with a mobile anchor node based on trilateration in wireless sensor networks, in: 2011 IEEE Global Telecommunications Conference - GLOBECOM 2011, 2011, pp. 1–6.

[16] F. B. Sorbelli, S. K. Das, C. M. Pinotti, S. Silvestri, Precise localization in sparse sensor networks using a drone with directional antennas, in: Proceedings of the 19th International Conference on Distributed Computing and Networking, ICDCN '18, ACM, New York, NY, USA, 2018, pp. 34:1–34:10.

[17] C. M. Pinotti, F. Betti Sorbelli, P. Perazzo, G. Dini, Localization with guaranteed bound on the position error using a drone, in: Proceedings of the 14th ACM International Symposium on Mobility Management and Wireless Access, MobiWac '16, ACM, New York, NY, USA, 2016, pp. 147–154.

[18] Enac, UAV regulations in Italy, `https://goo.gl/vgktKd` (2018 (accessed April 10, 2018)).

[19] G. Heredia, F. Caballero, I. Maza, L. Merino, A. Viguria, A. Ollero, Multi-unmanned aerial vehicle (uav) cooperative fault detection employing differential global positioning (dgps), inertial and vision sensors, Sensors 9 (9) (2009) 7566–7579.

[20] DecaWave, DecaWave Ltd: ScenSor SWM1000 Module, `http://www.decawave.com/products/dwm1000-module` (2018 (accessed April 10, 2018)).

[21] J.-S. Lee, Y.-W. Su, C.-C. Shen, A comparative study of wireless protocols: Bluetooth, uwb, zigbee, and wi-fi, in: Industrial Electronics Society, 2007. IECON 2007. 33rd Annual Conference of the IEEE, Ieee, 2007, pp. 46–51.

[22] A. Alarifi, A. S. Al-Salman, M. Alsaleh, A. Alnafessah, S. Alhadhrami, M. A. Al-Ammar, H. S. Al-Khalifa, Ultra wideband indoor positioning technologies: Analysis and recent advances, Sensors 16 (5) (2016) 707.

[23] F. B. Sorbelli, S. K. Das, C. M. Pinotti, S. Silvestri, On the accuracy of localizing terrestrial objects using drones, in: IEEE International Conference on Communications, ICC 2018, 2018, accepted for publication.

[24] H. T. Hu, F. C. Chen, Q. X. Chu, A compact directional slot antenna and its application in MIMO array, IEEE Transactions on Antennas and Propagation 64 (12) (2016) 5513–5517.

[25] 3D Robotics, Solo Specs: Just the facts, `https://news.3dr.com/solo-specs-just-the-facts` (2018 (accessed April 10, 2018)).