Set-Membership Approach to the Kidnapped Robot Problem

Benoît Desrochers[1,2], Simon Lacroix[1,3] and Luc Jaulin[2]

***Abstract*— This article depicts an algorithm which matches the output of a Lidar with an initial terrain model to estimate the absolute pose of a robot. Initial models do not perfectly fit the reality and the acquired data set can contain an unknown, and potentially large, proportion of outliers. We present an interval based algorithm that copes with such conditions, by matching the Lidar data with the terrain model in a robust manner. Experimental validations using different terrain model are reported to illustrate the performance of the method.**

I. Introduction

Metric maps of buildings, urban and natural environments are becoming widely available, and make map-based localization a key function to solve the overall robot localisation problem. By providing an "absolute" position estimate (actually relative to the map frame), map-based localization can indeed be exploited to solve the kidnapped robot problem, to palliate the inherent drift of dead-reckoning, to complement SLAM approaches, or in multi-robot teams where one robot localizes itself within a map built by an other.

a) Related work: The literature provides various approaches to map-based localization, which can be characterized by the way the acquired data are matched to the prior map (data association), and the algorithms that compute the position from the matches (estimation). Various map/data associations have been proposed, that either exploit skyline [1], [2], raw range data [3], [4], or detected features (*e.g.* reflective road markings [5]). As for estimation, particle filters are often used [6], [7]. One of their advantage is that they do not rely on explicit data association: only a likelihood measure is required between the acquired data and the initial map for a given position hypothesis, and the estimation scheme progressively discards wrong hypotheses. Yet this approach needs to be complemented with additional motion estimation techniques for the prediction step, and depending on the environment, can require lengthy displacements to provide a precise localisation estimate when starting without any prior information.

b) Motivations and approach: In most of environments, map-based localization approaches are challenged by the evolution of the environment, which generate numerous outliers, and hence considerably hinder the data association process or the likelihood measure of a position hypothesis. Our aim is to define an approach to estimate the robot position that is *robust* with respect to these changes. For that purpose, we rely on the following two choices:

- Exploit *geometric information*: the geometry of the environment is an intrinsic property that does not depend on illumination conditions, neither on the sensors that capture it and their viewpoints. Furthermore, most of initial maps are geometric, and in particular DEMs are easily built with aerial or ground means;
- Exploit interval analysis, that casts the localization problem into a set inversion problem. The literature has shown that interval analysis defines *robust* estimation solutions, while ensuring *integrity*: when an estimation is provided, the actual position is guaranteed to lie within the associated bounds [8], [9], [10], [11].

Yet, the potential large number of outliers, the facts that initial maps are always partial and that the initial position can be very poorly known challenge the existing set membership approaches. The contribution of this work is the proposal of an interval analysis algorithm that can cope with these problems. The approach is in particular able to reduce in a guaranteed manner initial position uncertainties as large as several tens of meters and degrees down to less than a meter and a few degrees, using a single depth image, and thus not requiring any robot motion.

c) Outline: The next section briefly recalls the basics of set inversion and the *GOMNE* algorithm, an essential contribution of the interval analysis literature to achieve robust set inversion in the presence of an unknown proportion of outliers. Our extension to this algorithm is presented in section III, and section IV presents its application to the robot localization problem using range data and initial DEMs. A discussion concludes the article.

II. Background On Interval Analysis For Estimation

This section recalls some basic tools of interval analysis (refer to [12] for an extensive description of the formalism), which will be extended in section III to solve our initial localization problem.

A. Intervals

An interval $[x]$ is a closed and connected subset of $\mathbb{R}$. A box $[\mathbf{x}]$ of $\mathbb{R}^n$ is a Cartesian product of n intervals. The set of all boxes is denoted $\mathbb{IR}^n$. The *width* of a box, denoted w, is the length along each dimension of its each interval.

B. Contractors

A *contractor* $\mathcal{C}$ is an operator from $\mathbb{IR}^n$ to $\mathbb{IR}^n$ such as:

$$\begin{array}{ll} \mathcal{C}([\mathbf{x}]) \subset [\mathbf{x}] & (contractance) \\ [\mathbf{x}] \subseteq [\mathbf{y}] \Rightarrow \mathcal{C}([\mathbf{x}]) \subseteq \mathcal{C}([\mathbf{y}]) & (monotonicity) \end{array} \quad (1)$$

[1]CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France
[2] ENSTA-Bretagne, LabSTICC, 2 rue François Verny, 29806 Brest
[3] Univ de Toulouse, LAAS, F-31400 Toulouse, France
benoit.desrochers@ensta-bretagne.org,
simon.lacroix@laas.fr,
luc.jaulin@ensta-bretagne.fr

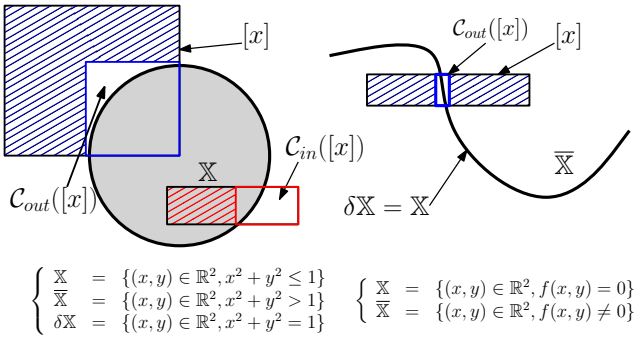


Fig. 1: The disk, described by an inequality, admits an inner contractor $\mathcal{C}_{in}$ and an outer contractor $\mathcal{C}_{out}$. On the contrary, only an outer contractor can be defined for the curve.

A set $\mathbb{X}$ is *consistent* with the contractor $\mathcal{C}$, if:

$$\forall [\mathbf{x}], \mathcal{C}([\mathbf{x}]) \cap \mathbb{X} = [\mathbf{x}] \cap \mathbb{X} \tag{2}$$

A contractor is *thin* if :

$$\forall \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \in \mathbb{X} \Leftrightarrow \mathcal{C}(\{\mathbf{x}\}) \neq \emptyset \tag{3}$$

where $\{\mathbf{x}\}$ is the degenerated box which contains as a single element the point $\mathbf{x}$.

Interval Arithmetics makes it possible to build efficient contractors associated with a set $\mathbb{X}$ defined by a set of constraints $\mathscr{C}$ such as equations or inequalities (see [12]). Figure 1 illustrates the notion of contractor for three sets: the contractor $\mathcal{C}_{out}$ associated with $\mathscr{C}$ removes part of an arbitrary initial box $[\mathbf{x}]$ which are not consistent with $\mathscr{C}$ without removing any feasible values. Such a contractor is an *outer* contractor. If $\mathbb{X}$ has a non empty volume, it is possible to define the *inner* contractor $\mathcal{C}_{in}$ which removes parts of $[\mathbf{x}]$ which belong to the $\mathbb{X}$ set.

C. Set Inversion

Let $\mathbf{f}$ be a function from $\mathbb{R}^n$ to $\mathbb{R}^m$ (possibly non-linear) and let $\mathbb{Y}$ be a subset of $\mathbb{R}^m$ such as for instance a sub-paving (finite union of non-overlapping boxes). The Set Inversion is the characterization of:

$$\mathbb{X} = \{\mathbf{x} \in \mathbb{R}^n | \mathbf{f}(\mathbf{x}) \in \mathbb{Y}\} = \mathbf{f}^{-1}(\mathbb{Y}) \tag{4}$$

For any $\mathbb{Y} \subset \mathbb{R}^m$ and for any function $\mathbf{f}$ admitting a convergent inclusion function $[\mathbf{f}](.)$, two regular sub-pavings $\mathbb{X}^-$ and $\mathbb{X}^+$ can be obtained with an arbitrary precision using the algorithm SIVIA (Set Inversion Via Interval Analysis) [13] such that:

$$\mathbb{X}^- \subset \mathbb{X} \subset \mathbb{X}^+ \tag{5}$$

The principle of SIVIA is to partition the search space into boxes and by applying contractors for the set to be characterized $\mathbb{X}$, and also for its complementary $\overline{\mathbb{X}}$.

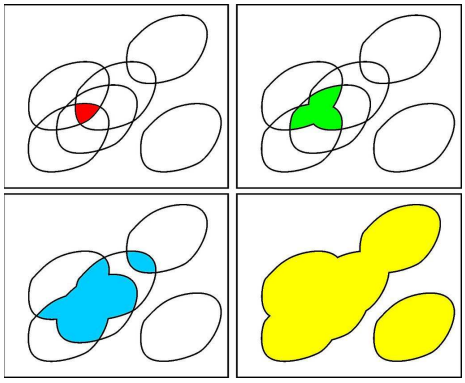


Fig. 2: q-relaxed intersection of 6 sets for q=2 (red), q=3 (green), q= 4 (blue) , q= 5 (yellow)

D. Relaxed Intersection

Parameters estimation using interval analysis is more efficient if the number of sets involved is greater than the number of unknowns. However the presence of outliers will yield an empty solution set. To solve this issue, the relaxed-intersection [14] has been introduced.

Consider N sets $\mathbb{X}_1, \ldots, \mathbb{X}_n$ of $\mathbb{IR}^n$. The $q-$relaxed intersection denoted by $\bigcap^{\{q\}} \mathbb{X}_i$ is the set of all $\mathbf{x} \in \mathbb{R}^n$ which belong to all $\mathbb{X}_i$'s, except q at most. Figure 2 illustrates this notion for N = 6 and q = 2, 3, 4.

Knowing q, the relaxed-intersection is mainly used inside the set inversion algorithm instead of the classical intersection to take into account at most q outliers.

E. Guaranteed Outlier Minimal Number Estimator

In actual cases, the percentage of outliers is not known and needs to be estimated concurrently with the parameters. The Guaranteed Outlier Minimal Number Estimator (*GOMNE*) algorithm, introduced in [15], has been designed for this purpose.

Let us take N sets $(\mathbb{X}_1, \ldots, \mathbb{X}_N)$ defined by constraints such as $f_i(\mathbf{x}) \in [\mathbf{y}_i]$. Let j be a cost function which returns the number of measurements inconsistent with $\mathbf{x}$.

$$\begin{array}{rccl} j: & \mathbb{R}^n & \longrightarrow & [\![1, N]\!] \\ & \mathbf{x} & \longmapsto & card\{i | f_i(\mathbf{x}) \notin [\mathbf{y}_i]\} \end{array} \tag{6}$$

The *GOMNE* aims at characterizing the set $\mathbb{S}^*$ of feasible parameters which minimizes the number of outliers:

$$\mathbb{S}^* = \arg\min_{\mathbf{x} \in \mathbb{R}^n} j(\mathbf{x}) \tag{7}$$

GOMNE runs a finite sequence of set inversions with different values of q in order to enclose $\mathbb{S}_{q^*}$ between two sub-pavings $\mathbb{S}^-_{q^*}$ and $\mathbb{S}^+_{q^*}$ in a guaranteed way. With an initial box $[\mathbf{x}_0]$, it starts with $q = 0$ and calculates $\mathbb{S}^-_q$ and $\mathbb{S}^+_q$ by running SIVIA($[\mathbf{x}_0]$,ϵ). If the outer sub-paving $\mathbb{S}^+_q$ is empty, there does not exist any solution consistent with at least $N-q$ data. The number of outliers q is incremented and SIVIA is run again. If $\mathbb{S}^+_q$ is not empty and $\mathbb{S}^-_q$ is, a smaller value of ϵ is used to find a non empty inner set or an empty outer set.

In the first case, the algorithm ends and $q^* = q$, otherwise SIVIA is run again with a greater value of q.

III. Outer-GOMNE

When the set to be characterized has an empty volume (as it is the case for our localisation problem, see section IV), no inner contractor will be able to contract any boxes (except the degenerated boxes which are singletons). In the case, GOMNE will never terminate because the stopping condition ($\mathbb{S}^- \neq \emptyset$) can not be satisfied. To cope with this lack of constraint, we propose to use the local information given by j (see equation 6) to find the smallest upper bound of q^* as possible. The outer contractor will remove inconsistent parts of the initial space while a local method looks for q^*.

A. Principle

Let us take a set of N measurements, each of them defining a set $\mathbb{X}_i$. Let $\mathcal{C}^i_{out}$ be a *thin* outer contractor associated with the set $\mathbb{X}_i$. The method is based on the following proposition:

Proposition: If j is the function defined by (6), we have

$$j(\mathbf{x}) = \sum_{i=1}^{N} \mu_i(\mathbf{x}) \text{ where } \mu_i(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathcal{C}^i_{out}(\{\mathbf{x}\}) = \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

Proof: it is a direct consequence of the thin property of $\mathcal{C}^i_{out} : \mu_i(\mathbf{x}) = 1 \Leftrightarrow f_i(\mathbf{x}) \notin [\mathbf{y}_i] \Leftrightarrow \mathbf{x} \notin \mathbb{X}_i$.

Note that properties of j such as continuity, variations, monotonicity are unknown: j is not differentiable and it can only be evaluated at a given value.

Properties: Based on the construction of j and on the monotonicity of the relaxed intersection we have the three following relatives:

$$\begin{array}{ll} \text{(i)} & \mathbb{S}^* \neq \emptyset \Leftrightarrow \exists \mathbf{x} \in \mathbb{R}^n \ , \ j(\mathbf{x}) = q^* \\ \text{(ii)} & \forall \mathbf{x} \in \mathbb{R}^n, \mathbb{S}_{q^*} \subseteq \mathbb{S}_{j(\mathbf{x})} \\ \text{(iii)} & \forall \mathbf{x} \notin \mathbb{S}^*, j(\mathbf{x}) > q^* \end{array} \tag{9}$$

With equation (i), the minimum of j can always be found if the search algorithm has no time limit. Otherwise, with the property (ii), it can be stopped at any time and the guarantee is preserved. With the last property (iii), algorithms can be run safely with any $\mathbf{x} \in \mathbb{R}^n$.

To find the minimal value of the function j in a given sub-paving $\mathbb{A}$, several strategies can be used. A trivial way is to randomly test N points in $\mathbb{A}$ and keep the minimal value. The probability of finding the minimum increases when N tend towards infinity or when the volume of $\mathbb{A}$ approaches towards 0. A smarter way is to consider j as a likelihood function and use the particle filter framework to estimate $\mathbb{S}^*$. Alternatively, the issue of finding the best value of j could be seen as a black-box optimization problem because no information are available on j. We choose here to use the NOMAD library [16] which is a C++ implementation of the Mesh Adaptive Direct Search (MADS) algorithm [17] designed for constrained optimization of black-box functions.

B. Algorithm

The initial *GOMNE* algorithm is modified in order to take into account the new way to estimate q^*. At the beginning, $[q] = [q^-, q^+]$ is unknown and ranges between 0 and $N-1$. Similarly to *GOMNE*, a set inversion algorithm is applied. If the resulting sub-paving $\mathbb{S}^+$ is empty, there are at least q outliers and $q^- = q + 1$; else the minimization algorithm is run inside $\mathbb{S}^+$ to find a better value for q^+. The algorithm stops when $q^- \geq q^+$, and then $q^* = q^+$.

Remarks:

- The returned value is not always optimal and depends on the result of the minimization. However, if $\mathbb{S}^+$ is small, the initial search space is also small and an optimal value is found quickly.
- After the optimization, if no improvement of the value of q is found (line 8), a smaller accuracy (ϵ) is used to reduce the size of boxes which composed the $\mathbb{S}^+$ and then reduce the initial search space.
- When the number of outliers is greater than 50% testing iteratively each value of q could be very slow. To speed up the algorithm, q can be incremented by Δq and the best value which satisfies $q^- >= q^+$ will be found quickly but only satisfies $q^+ - q^* \leq \Delta q$.

Algorithm 1: Outer-GOMNE(in: X_0 , $\epsilon_0, \epsilon_{lim}$, out: q^+)

```
1  q⁻ := 0 , q⁺ := N − 1, ε := ε0;
2  while (q⁺ > q⁻) do
3  |   S⁺ := SIVIA([x0], ε);
4  |   if (S⁺ = ∅ or ε < ε_lim) then
5  |   |   q⁻ := q⁻ + 1, ε := ε0
6  |   else
7  |   |   q := min_{x∈S⁺} j(x);
8  |   |   if (q < q⁺) then
9  |   |   |   q⁺ := q
10 |   |   else
11 |   |   |   ε := ε/2
```

Results of this algorithm with a Lidar on a DEM are shown in the next section. Further analysis of the behavior of the algorithm can be found in [18], where extensive simulations in a 2D simulated worlds are conducted.

IV. Localization in a Digital Elevation Map

DEM's represent a surface $z = f(x, y)$ on a regular Cartesian grid where each cell contains the value of the elevation z. DEM's are widely used, either in geographic data systems because they present a good compromise between expressiveness, simplicity and compactness[1], or in robotics, where they are easily built from range data.

If a DEM is an adequate structure to represent surfaces, it fails to represent verticals and overhangs, and this has numerous consequences, especially when it comes to associate

[1]For instance, most commercial aerial mapping systems generate DEMs along with orthoimages

range data acquired by the robot to the map. Furthermore, the presence of unmapped elements in the environment, and the fact that some objects may only be partially mapped make the localization challenging. This section presents how *Outer-GOMNE* can cope with these issues, using Lidar data acquired by the robot.

A. Initial Models

1) Map Structure Issues: With a DEM, the space can be split into two sets: the one over the surface $(z - f(x, y) > 0)$ and the one below $(z - f(x, y) < 0)$. In such cases, an inner and an outer contractor can be defined, and the classical *GOMNE* approach can be used to cope with the presence of outliers. For instance, if a DEM is built from an Unmanned Aerial Vehicle (UAV) using a downwards oriented sensor (see figure 3), range data acquired by other UAVs using a downwards oriented sensor can be well matched with the DEM structure. But on the contrary, range data acquired by a ground robot will hardly match such a map: *e.g.* the tree canopy is mapped by the UAV, whereas the ground robot only perceives tree trunks and lower branches and leaves. From the point of view of a UAV, the representation as a surface (convex world) is a good approximation. However from the point of view of the ground robot, the elevation between two adjacent cells is assumed to be continuous, which leads to a poor representation of verticals and overhangs.

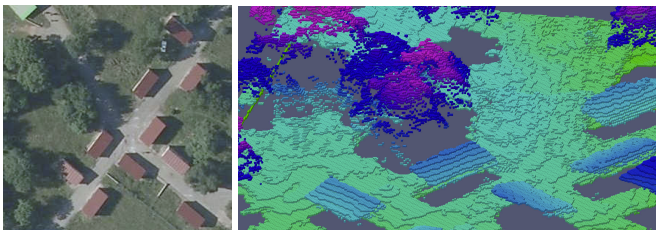


Fig. 3: Aerial orthoimage of a 80m x 80m test site, and DEM built from a UAV with a downwards looking Lidar. Roofs and tree canopy are disconnected from the ground surface.

As a compromise, the DEM can be voxelized, by filling the gaps between two different elevations of neighboring cells. The figure 4 shows the resulting environment model after applying this process to the DEM of figure 3. Note that this representation also allows the use of an image contractor extended to the 3D case. But if this copes for verticals such as walls, it also clearly introduces erroneous information: the extension to the ground of the tree canopy does not fit the reality and hides the tree trunks for instance.

To better solve this issue, a volumetric representation using a three-dimensional voxel grid can be used: non surfacic elements are then accurately represented. However, the building of such models has a high computational cost and requires a lot of memory. And still, some objects may not be entirely mapped in the model: *e.g.* sometimes only parts of buildings, cars, trees, are modeled and the resulting model does not define closed connected components, as can be seen in the model of figure 5, where only walls perceived

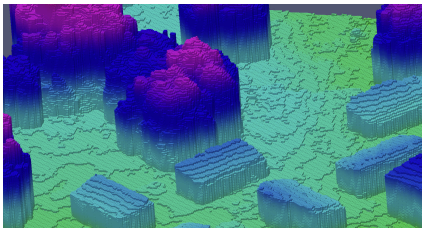


Fig. 4: View of the DEM of figure 3 after its "voxelisation".

by the robot are modeled – and not the entire buildings. The hypothesis of a convex world then does not hold, and no inner contractor are available: in such a situation, *Outer-GOMNE* needs to be used.

2) Test Maps: For our localisation tests, three different models of the same area (training camp of Caylus denoted Caylus-1, figure 3) are used. The first model, denoted "atLaas", is built from a set of 3D scans acquired by a Velodyne Lidar mounted on a ground robot (figure 5). The second and third models are built from data acquired by an UAV equipped with a camera (6a) and a Lidar (figure 6b), and are respectively denoted "UAV-Vision" and "UAV-Lidar". These three models are "voxelized".

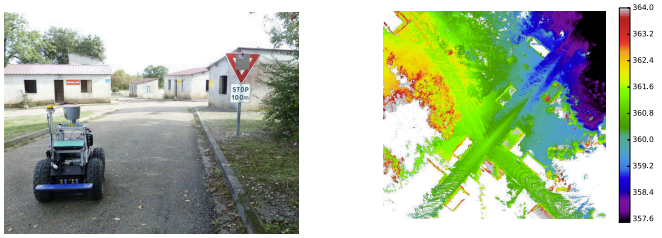
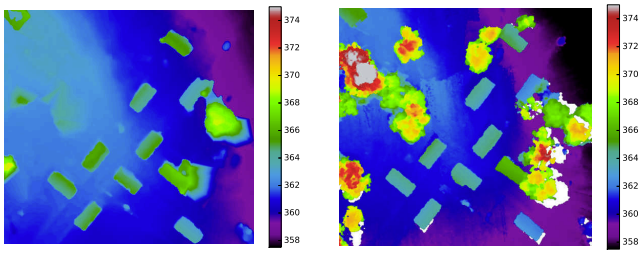


Fig. 5: The ground robot with a Velodyne Lidar in Caylus-1, and the corresponding "atLaas" DEM (0.1m resolution). Unseen areas, like roof, are represented in white.

B. Practical implementation

The localization problem is formalized as a robust set inversion. Let $\mathbb{M}$ be the set of all occupied voxels of



(a) "UAV-Vision" DEM built from images (0.04m resolution)

(b) "UAV-Lidar" DEM built from Lidar data (0.2m resolution)

Fig. 6: DEMs of Caylus-1 generated by UAV data. Note that the DEM built from images contains gross errors, due to the absence of visual features matches: the elevation has been interpolated by the photogrammetric algorithms to the ground elevation for most of the trees.

the initial model. Let also $\mathbf{y}^i_{B_s} = (x_i, y_i, z_i)$ be a point acquired by the sensor (a Velodyne HDL-64) in its own frame B_s. In our context, $\mathbf{y}^i_{B_s}$ is assumed to belong to the box $[\mathbf{y}^i_{B_s}]$, centered on $\mathbf{y}^i_{B_s}$ and inflated by 0.01ρ where ρ is the distance to the sensor. The goal is to estimate the parameters $\mathbf{x} = (p_x, p_y, p_z, \varphi, \theta, \psi)^T$ of the transformation matrix $\mathcal{T}_{B_0 \to B_s}(\mathbf{x})$ between the global frame B_0 and the sensor frame B_s. If $\mathbf{y}^i_{B_0}$ is an inlier, it must belong to a voxel of the map. This constraint is described by:

$$\begin{cases} \mathbf{y}^i_{B_0} = \mathcal{T}_{B_0 \to B_s}(\mathbf{x}).\mathbf{y}^i_{B_s} \\ \mathbf{y}^i_{B_0} \in \mathbb{M} \end{cases} \tag{10}$$

The solution set which minimize the number of outliers is defined by:

$$\mathbb{S}^* = \bigcap^{\{q^*\}} \mathcal{T}^{-1}_{B_0 \to B_s}(\mathbb{M}) \tag{11}$$

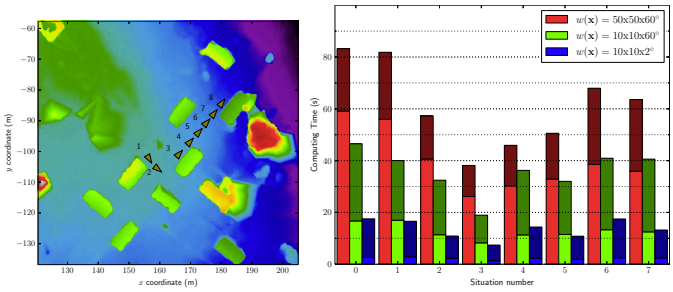
where *Outer-GOMNE* can be used to concurrently determine $\mathbf{x}$ and q^*. The constraint $\mathbf{y}^i_{B_0} \in \mathbb{M}$ is implemented using a voxel grid extension of the image contractor [19] applied on the generalized notion of integral images [20].
The estimation of $\mathbf{x}$ can be split into two sub-problems: estimate $\mathbf{x}_1 = (p_x, p_y, \varphi)^T$ which uses mainly information provided by the relief (walls, trees, buildings, slopes, ...) and $\mathbf{x}_2 = (p_z, \theta, \phi)^T$ which is based on the perception of the ground. In our experiments we only deal only with the estimation of $\mathbf{x}_1$ and input data are filtered in this regard (θ and ϕ are well observed by an IMU at rest, and p_z is trivially deduced form the estimation of (p_x, p_y) and the DEM). For that purpose, over the 100 000 points returned by the sensor, a ground segmentation algorithm is used to keep only those which are relevant for the localization.

C. Results

The approach has been evaluated in different positions of the robot using two ground Lidar data sets acquired in two different days, with ground truth positions provided by a centimeter accuracy RTK-GPS. One data-set has been used to generate the "atLaas" DEM (figure 5), while the second data-set is used to test the localisation algorithm in this DEM and the two "UAV-Vision" and "UAV-Lidar" DEMs derived from UAV data.

Localisation results on the "atLaas" DEM for 8 robot positions are shown in figure 7a, execution time with different sizes of the initial box are given in figure 7b (measured on a Intel Core i5 CPU at 2.50GHz), and the characteristics of the resulting sub-paving are given in table I. In each case, in accordance with the theory, the integrity of the result is satisfied and the size of the box which encloses the resulting sub-paving is smaller than one meter. Table I also shows the errors err_{xy} and err_θ, which have been computed by comparing the ground truth with the center of the box that bounds the resulting sub-paving.

Computing times obviously depend on the initial information on the robot position (box size), and the complexity of the algorithm is linear with respect to the number of measurements used. Using less measurements would speed up the algorithm and especially the optimization part – yet a way to select relevant measurements has to be defined.



(a) Positions estimated for 8 positions.

(b) Computation time with different sizes of the initial boxes.

Fig. 7: Localisation results using the "atLaas" DEM as the initial map (here shown on the "UAV-Vision" DEM), and execution times as a function of the size of the initial box. Light colored bars correspond to the time needed by the set inversion algorithm.

#	Nb points	% of outliers	$w([x])$ m	$w([y])$ m	$w([\theta])$ $mrad$	err_{xy} m	err_θ $mrad$
1	1174	9 %	0.78	0.81	82.3	0.20	9.8
2	1154	2 %	0.41	0.55	50.7	0.20	3.6
3	786	4 %	0.56	0.55	53.9	0.16	18.2
4	738	11 %	0.67	0.79	56.3	0.03	6.3
5	805	9 %	0.80	0.75	67.0	0.09	1.0
6	912	9 %	0.81	0.67	56.3	0.20	6.3
7	1038	8 %	0.76	0.75	80.0	0.10	18.2
8	851	5 %	0.63	0.68	67.0	0.28	24.8

TABLE I: Characteristics of the resulting sub-paving (with a large initial box of 50m x 50m x 60°) for the 8 position estimates shown figure 7.

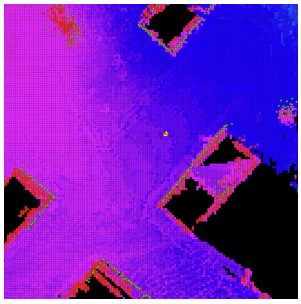
Table II and figure 8 illustrate the localisation results of one other configuration with the three initial DEMs considered. In figure 8, the resulting sub-paving is displayed in yellow, the red triangle is the estimated position of the robot, and the range data that match the DEM is superimposed in green, while the estimated data outliers are in red. A straightforward observation is that using the aerial DEMs, Outer-GNOME can deal with the numerous model errors, that generate up to 59% outliers.

Figure 8a shows the final situation when the "atLaas" DEM is used. As for the former trajectory, the robot is very accurately localized: the range data are indeed acquired in conditions close to the acquisition conditions to built the model, and contain less than 10% of outliers. With the "UAV-Lidar" DEM, *Outer-GOMNE* succeeds to find the pose with a much higher proportion of outliers (figure 8b).

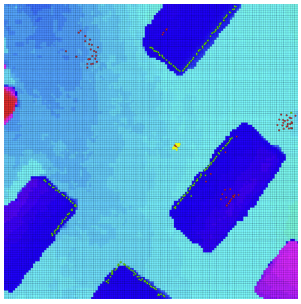
The "UAV-Vision" DEM unfortunately contains a scale error and it is nearly impossible for the scan to match two parallel walls in the same time. As a result, the estimated sub-paving is made of three disconnected components, which correspond to different data pairing. Figure 8d show the data points reprojected considering one of this component: they are well aligned with walls of the building on the top, but

DEM	atLaas	UAV-Vision	UAV-Lidar
$N_{outliers}$	10 (3%)	158 (54%)	102(34%)
time(s)	7.68	105	8.38
$w([p_x])\ m$	0.36	3.67	0.81
$w([p_y])\ m$	0.37	4.54	0.76
$w([\theta])\ rad$	0.060	0.04	0.035

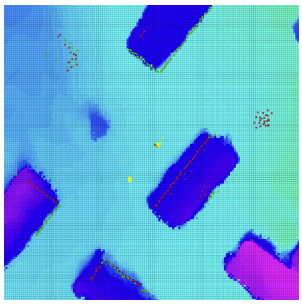
TABLE II: Results of *Outer-GOMNE* for one position estimated on three different DEMs using 293 measurements.



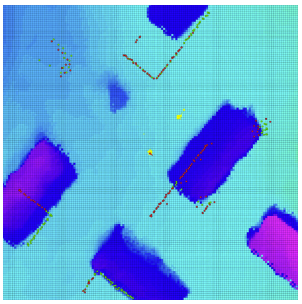
(a) Situation B on atLaas DEM



(b) Situation A on UAV-Lidar DEM



(c) Situation B on UAV-Vision DEM – first component of the sub-paving



(d) Situation B on UAV-Vision DEM – second component of the sub-paving

Fig. 8: Localisation results on the 3 different DEMs considered.

other points are inside the middle building. Consequently, the number of outliers increases. Another component of the sub-paving, for which the data points are shown in figure 8c, corresponds to an unfeasible pose because the constraint used allows scans to pass through walls.

V. Summary and Hints for Future Work

One of the main issue of map-based localization is the presence of outliers, that mainly comes from changes in the scene and limitations of the map structure. Being robust with respect to these outliers is of course essential: set membership approaches to this problem have shown to have such a robustness, the *GOMNE* algorithm being able to estimate the position parameters even in the case of an unknown bound on the outliers.

However the initial *GOMNE* algorithm requires an inner contractor, which can not be provided when the initial model is partial and exhibits non-closed components – which is the case in most operational applications. To cope with this, we proposed *Outer-GOMNE*, that combines a set inversion algorithm with an optimization method and that does not require an inner contractor. Experiments showed that *Outer-GOMNE* is robust to gross map errors and able to face complex situations.

The algorithm is not designed for continuous localization: it rather aims at providing an initial and guaranteed set of feasible positions. Interestingly, it does not require any movement of the platform. It can also be used, for instance, to focus a particle filter on a restricted portion of the space, which is guaranteed to contain the position of the robot. Further work is required to quantitatively assess the benefits of this integration of an interval approach with a Monte Carlo localization method could bring in terms of convergence time, computation time, and robot motions required.

References

[1] S. Ramalingam, S. Bouaziz, P. Sturm, and M. Brand, "SKYLINE2GPS: Localization in Urban Canyons using Omni-Skylines," in *IROS*, 2010.

[2] F. Cozman, E. Krotkov, and C. E. Guestrin, "Outdoor visual position estimation for planetary rovers," *Autonomous Robots*, vol. 9, pp. 135–150, 2000.

[3] N. Vandapel, R. R. Donamukkala, and M. Hebert, "Unmanned ground vehicle navigation using aerial ladar data," *The International Journal of Robotics Research*, vol. 25, no. 1, pp. 31–51, 2006.

[4] P. J. F. Carle, P. T. Furgale, and T. D. Barfoot, "Long-Range Rover Localization by Matching LIDAR Scans to Orbital Elevation Maps," *Journal of Field Robotics*, vol. 27, no. 3, pp. 344–370, 2010.

[5] J. Levinson, M. Montemerlo, and S. Thrun, "Map-Based Precision Vehicle Localization in Urban Environments," in *RSS*, 2007.

[6] D. Fox, S. Thrun, F. Dellaert, and W. Burgard, "Particle filters for mobile robot localization," in *Sequential Monte Carlo Methods in Practice*, A. Doucet, N. de Freitas, and N. Gordon, Eds. New York: Springer Verlag, 2000.

[7] R. Zapata, L. Zhang, and P. Lepinay, "Self-adaptive monte carlo localization for mobile robots using range finders," *Robotica*, 2012.

[8] D. Meizel, O. Lévêque, L. Jaulin, and E. Walter, "Initial localization by set inversion," *IEEE transactions on robotics and Automation*, 2002.

[9] V. Drevelle and P. Bonnifait, "Reliable positioning domain computation for urban navigation," *IEEE Intelligent Transportation Systems Magazine*, vol. 5, no. 3, pp. 21–29, 2013.

[10] E. Colle and Galerne, "Mobile robot localization by multiangulation using set inversion," *Robotics and Autonomous Systems*, 2013.

[11] R. Guyonneau, S. Lagrange, L. Hardouin, and P. Lucidarme, "The kidnapping problem of mobile robots : A set membership approach," in *7th National Conference on Control Architectures of Robots*, 2012.

[12] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied Interval Analysis*. Springer, 2001.

[13] L. Jaulin and E. Walter, "Set inversion via interval analysis for nonlinear bounded-error estimation," *Automatica*, 1993.

[14] L. Jaulin, "Robust set membership state estimation ; application to underwater robotics," *Automatica*, vol. 45, no. 1, pp. 202–206, 2009.

[15] L. Jaulin, M. Kieffer, E. Walter, and D. Meizel, "Guaranteed robust nonlinear estimation with application to robot localization," *IEEE Transactions on SMC, Part C*, nov 2002.

[16] M. Abramson, C. Audet, G. Couture, J. Dennis, Jr., S. Le Digabel, and C. Tribes, "The NOMAD project," Software available at http://www.gerad.ca/nomad.

[17] C. Audet and J. Dennis, "A progressive barrier for derivative-free nonlinear programming," *SIAM Journal on Optimization*, vol. 20, no. 1, pp. 445–472, 2009.

[18] B. Desrochers, "Set-Membership Approach to Map-Based Localization," LAAS-CNRS ; ENSTA-Bretagne, Tech. Rep., 2014. [Online]. Available: http://hal.archives-ouvertes.fr/hal-01065126

[19] J. Sliwka, "Using set membership methods for robust underwater robot localization," Ph.D. dissertation, Ensta Bretagne, 2011.

[20] E. Tapia, "A note on the computation of high-dimensional integral images," *Pattern Recognition Letters*, vol. 32, no. 2, pp. 197 – 201, 2011.