# Image-based UAV localization using Interval Methods

Ide-Flore Kenmogne, Vincent Drevelle and Eric Marchand

***Abstract*— This paper proposes an image-based localization method that enables to estimate a bounded domain of the pose of an unmanned aerial vehicle (UAV) from uncertain measurements of known landmarks in the image. The approach computes a domain that should contain the actual robot pose, assuming bounded image measurement errors and landmark position uncertainty. It relies on interval analysis and constraint propagation techniques to rigorously back-propagate the errors through the non-linear observation model. Attitude information from onboard sensors is merged with image observations to reduce the pose uncertainty domain, along with prediction based on velocity measurements. As tracking landmarks in the image is prone to errors, the proposed method also enable fault detection from measurement inconsistencies. This method is tested using a quadcopter UAV with an onboard camera.**

## I. Introduction

To navigate and accomplish their tasks, robots not only need to locate themselves with respect to the environment but also need to have confidence information about their position. In the case of Unmanned Aerial Vehicles (UAVs), the standard solution consists in using GPS, INS, barometer or Compass measurements. Yet, this solution is not appropriate in difficult environment such as, eg, indoors areas, proximity to large buildings where GPS signals are blocked or unreliable, or near steel structures that jeopardize compass readings. Our goal here is to provide a reliable pose confidence domain; a box in which we are sure the robot is located. In other words, we wish to compute a "safety area" around the robot that can be considered by the controller in order to avoid collisions with other robots or objects present in the navigation environment.

A lot of commercially available UAVs carry an onboard video camera, mainly used to provide a visual feedback to the pilot and to shoot aerial videos. This paper thus focuses on using an onboard camera in order to enhance the robot localization. Using a image-based primary positioning system enables to overcome GPS and compass unreliability in difficult environments. Image-based visual servoing [1], [2] is a way to control robots in such environments. In computer vision, many solutions to pose estimation from a set of known landmarks (see, eg, [3]–[5] for a survey) exist but classically provide a punctual estimate of the robot location. Interval analysis is a powerful tool for rigorous uncertainty propagation which has been considered GPS position uncertainty domain computation [6] and for 3-D vision application [7].

In this paper, we aim at characterizing a domain that contains the pose (6-DOF position and attitude) of a robot equipped with a camera and proprioceptive sensors, by using interval-based constraints propagation techniques. More precisely, to quantify the robot pose uncertainty, we propose an interval-based set-membership approach [8], which computes over time a bounding box of the pose of the robot, taking image measurements and landmark positions uncertainties into account.

The contributions of the paper are:

- computation of a domain containing the robot pose, assuming bounded-error image measurements of known landmarks, and fusion with attitude and velocity sensors
- landmark tracking failure detection

This paper is structured as follow: In Section II we will first give a statement of the pose estimation problem and show the solutions that already exists in the literature, and a short recall of how uncertainty can be propagated. Section III will introduce interval analysis as a tool for bounded-error set-membership estimation. We will present the proposed method in Section IV based on the notion of set inversion and contractors. We will finish this article by presenting the results obtained with real data obtained from a quadcopter in Section V.

## II. Classical Approaches for Pose Estimation

This section is devoted to the description of *pose estimation* problem which consist in computing the position and orientation of a calibrated camera with respect to a given set of correspondences between 3D features and their projection in the images plane [4], [5]. Firstly, theoretical aspects of the problem are recalled and linked to robot pose estimation. Then, a short survey of the existing solutions to compute a point pose estimate is introduced and finally classical approaches to quantify uncertainty are reviewed.

### A. Problem statement

Considering the movement of a camera in a world (as depicted in Fig. 1), we can define the camera frame $\mathcal{F}_c$ attached to the camera optical center and the world frame $\mathcal{F}_w$ attached to the origin of a local coordinate system. In this context, the pose estimation problem rely on determining the transformation between the frames $\mathcal{F}_w$ and $\mathcal{F}_c$. In other words, determine the position of the camera in the world frame. Lets denote ${}^c\mathbf{T}_w$ the transformation from $\mathcal{F}_w$ to $\mathcal{F}_c$; it is defined as follows:

Ide-Flore Kenmogne is with Inria Rennes-Bretagne Atlantique, IRISA, Rennes, France ide-flore.kenmogne@inria.fr

Vincent Drevelle and Eric Marchand are with Université de Rennes 1, IRISA, Inria Rennes-Bretagne Atlantique, Rennes, France {vincent.drevelle, eric.marchand}@irisa.fr.

$$ {}^{c}\mathbf{T}_w = \begin{pmatrix} {}^{c}\mathbf{R}_w & {}^{c}\mathbf{t}_w \\ \mathbf{0}_{3\times 1} & 1 \end{pmatrix} \tag{1} $$

where ${}^{c}\mathbf{R}_w$ and ${}^{c}\mathbf{t}_w$ are respectively the rotation matrix (depending on the camera's orientation) and translation vector that represents the position and attitude of the camera in the world frame. To solve this problem, the perspective projection equation of a 3-D point (in the world frame) on the image frame (a 2-D point) is used. The pinhole camera model is expressed as follows:

$$ \bar{\mathbf{x}} = \mathbf{K}\,\mathbf{\Pi}\,{}^{c}\mathbf{T}_w\,{}^{w}\mathbf{X} \tag{2} $$

where :

- ${}^{w}\mathbf{X} = (X, Y, Z, 1)^\top$ the homogeneous coordinates of a 3-D point in the world frame;
- $\bar{\mathbf{x}} = (u, v, 1)^T$ are the homogeneous coordinates (in pixel) of the projection of ${}^{w}\mathbf{X}$ in the image;
- $\mathbf{K}$ is the camera intrinsic parameters matrix given by:

$$ \mathbf{K} = \begin{pmatrix} p_x & 0 & u_0 \\ 0 & p_y & v_0 \\ 0 & 0 & 1 \end{pmatrix} $$

  where $p_x$ and $p_y$ are the meter/pixel scale factor and $u_0$ and $v_0$ the image principal point's coordinates;
- $\mathbf{\Pi}$ is the perspective projection matrix given by:

$$ \mathbf{\Pi} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} $$

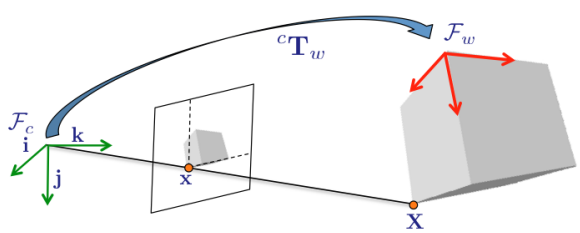


Fig. 1. Perspective projection of an object in the image plane

The intrinsic parameters can be easily obtained through an off-line calibration step (e.g. [9]). Therefore, when considering a localization, we shall consider image coordinates expressed in the normalized metric space $\mathbf{x} = \mathbf{K}^{-1}\bar{\mathbf{x}}$. Let us note that we consider here only a pure perspective projection model but it is clear that any model with distortion can be easily considered and handled. From now, we will always consider that the camera is calibrated and that the coordinates are expressed in the normalized space.

If we have $N$ points ${}^{w}\mathbf{X}_i, i = 1..N$ whose coordinates expressed in $\mathcal{F}_w$ are given by ${}^{w}\mathbf{X}_i = (X_i, Y_i, Z_i, 1)^\top$, the projection $\mathbf{x}_i = (x_i, y_i, 1)^\top$ of these points in the image plane is then given by: $\mathbf{x}_i = \Pi\,{}^{c}\mathbf{T}_w\,{}^{w}\mathbf{X}_i$. Knowing 2D-3D point correspondences, $\mathbf{x}_i$ and ${}^{w}\mathbf{X}_i$, pose estimation consists in solving the system of equations for ${}^{c}\mathbf{T}_w$. This is an inverse problem that is known as the Perspective from $N$ Points problem or P$n$P (Perspective-$n$-point).

The problem stated above is the camera pose estimate while in this article we aim at estimating a robot's pose. If we consider a camera mounted on a robot, a third frame has now to be taken in account which is the robot frame denoted by $\mathcal{F}_r$. In this case, the problem consists in determining the transformation ${}^{r}\mathbf{T}_w$ that fully defines the position of $\mathcal{F}_r$ in $\mathcal{F}_w$. Knowing ${}^{c}\mathbf{T}_r$ from calibration, the world-to-camera transformation is expressed by ${}^{c}\mathbf{T}_w = {}^{c}\mathbf{T}_r\,{}^{r}\mathbf{T}_w(q)$.

### B. Solving the PnP

Many solutions exist to solve the P$n$P problem. As far as the P3P is concerned, the points depth are first estimated by solving a fourth order polynomial equation [10]. Once the three points coordinates are known in the camera frame, the second step consists in estimating the rigid transformation ${}^{c}\mathbf{T}_w$ that maps the coordinates expressed in the camera frame to the coordinates expressed in the world frame.

Other PnP (with $n > 3$) methods exist and are generally adopted due to the fact that more points increase pose accuracy. The Direct Linear Transform (DLT) is certainly the oldest one [4] nevertheless, being over-parameterized, this solution is very sensitive to noise and a solution that explicitly considers the non-linear constraints of the system should be preferred. An alternative and very elegant solution, which takes these non-linear constraints into account, has been proposed in [11]. When complexity is of interest, non-iterative P$n$P algorithms with a linear complexity have been proposed. A first accurate $O(n)$ solution to the P$n$P was EP$n$P [12]. Later, other $O(n)$ solutions such as OP$n$P, GP$n$P, UP$n$P were proposed.

The "gold-standard" solution to the P$n$P consists in estimating the six parameters of the transformation ${}^{c}\mathbf{T}_w$ by minimizing the norm of the reprojection error using a non-linear minimization approach such as a Gauss-Newton of a Levenberg-Marquardt technique. Denoting $\mathbf{q} \in se(3)$ a minimal representation of ${}^{c}\mathbf{T}_w$ ($\mathbf{q} = ({}^{c}\mathbf{t}_w, \theta\mathbf{u})^\top$ where $\theta$ and $\mathbf{u}$ are the angle and the axis of the rotation ${}^{c}\mathbf{R}_w$), the problem can be formulated as:

$$ \widehat{\mathbf{q}} = \underset{\mathbf{q}}{\operatorname{argmin}} \sum_{i=1}^{N} d\big(\mathbf{x}_i, \Pi\,{}^{c}\mathbf{T}_w\,{}^{w}\mathbf{X}_i\big)^2 \tag{3} $$

where $d(\mathbf{x}, \mathbf{x}')$ is the Euclidean distance between two points $\mathbf{x}$ and $\mathbf{x}'$. The solution of this problem relies on an iterative minimization process such as a Gauss-Newton method.

We will see that our approach to pose estimation is very different and does not rely on the direct computation or estimation of the pose by solving equation (3).

### C. Uncertainties propagation

When estimating the camera or robot pose from image measurements, a common concern is to quantify the uncertainty of the estimate. Measurements are generally noisy and/or biased, and the observation model and its parameters are also tainted with uncertainty. Quantifying uncertainty usually consists in back-propagating observation error through the uncertain model to get an estimate of the uncertainty on the estimated quantity.

Probabilistic approaches are classically considered for this purpose: simulation-based techniques like Monte Carlo [13] or importance sampling [14] can deal with a large class of problems but don't scale well with problem dimension; methods based on Taylor series, like first-order second-moment (FOSM) are well suited when the nonlinearity is moderate over the range of the considered uncertainties, which often restricts them to small uncertainties. Other methods like Karhunen-Loeve expansion can be employed for a better handling of nonlinearity (see [15] for a more detailed comparative study of theses methods).

Using probabilistic approaches generally requires prior knowledge about the error distribution. A common assumption when using FOSM for propagation is that the error vector follows a centered multivariate normal distribution of known covariance matrix. In the lack of such information, the independent and identically distributed errors assumption is often taken.

Non-probabilistic approaches like interval analysis [16] or fuzzy theory [17] rely on different representations of the uncertainty. Interval analysis assumes that the errors are bounded, and that these bounds are known. It enables to represent and propagate worst-case uncertainty conditions, thus providing bounds on the estimation error. The only assumption made about the error probability distribution is that its support is bounded.

## III. Interval Analysis

This section presents basic concepts and tools of *Interval Analysis*, which is a non-probabilistic tool for bounded error estimation. It puts an emphasis on the use of Contractor Programming and Set Inversion via Interval Analysis (*SIVIA*) to perform set-membership estimation, i.e computing the set of solutions that are compatible with the measurements and their uncertainty.

### A. Interval Analysis

Interval analysis relates to computations with intervals and their multidimensional extension which are *interval vectors* or *boxes*. Unlike exact representation of sets, *intervals* and *boxes* are easy to manipulate. The set of real intervals is denoted $\mathbb{IR}$ and the set of n-dimensional boxes is $\mathbb{IR}^n$.

Like in [8], in this article we denote an interval or a box $[x] = [\underline{x}, \overline{x}]$ with the bracket notation, where $\underline{x}$ and $\overline{x}$ denote the lower and upper bounds of $[x]$.

The width of an interval $[x]$ is $\overline{x} - \underline{x}$ and the width of a box is the largest width of its interval components.

Interval arithmetic enables computations with intervals thanks to the interval extension of classical real arithmetic operators $+, -, \times$ and $\div$. For non-empty closed intervals, $[x] + [y] = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$ and $[x] - [y] = [\underline{x} - \overline{y}, \overline{x} - \underline{y}]$. Let us show an example:

$[1, 3] + [0, 5] = [1 + 0, 3 + 5] = [1, 8]$

$[0, 3] - [0, 3] = [0 - 3, 3 - 0] = [-3, 3]$

Similarly, elementary functions ($sin$, $cos$, $exp$, etc.) can be extended to intervals. This is done by returning the smallest interval that covers the range of the input through the function. The image of a box by a $\mathbb{R}^n \longrightarrow \mathbb{R}^m$ is generally not itself a box, but an arbitrary set. This is solved using *inclusion functions*.

Let $[f]$, an interval function from $\mathbb{IR}^n$ to $\mathbb{IR}^m$. $[f]$ is an *inclusion function* for $f$ if the image of $[x]$ by $f$ is included in the image of $[x]$ by $[f]$, meaning that :

$$\forall [x] \in \mathbb{IR}^n, f([x]) \subset [f]([x]). \tag{4}$$

If an inclusion function $[f^*]$ for a function $f$ returns the smallest box that contains $f(x)$ i.e.$[f]([x]) = f([x])$, then this inclusion function is *minimal* and is the *interval hull* of $f([x])$ [8]. The *natural inclusion function* for $f$ is obtained by replacing each operator in the expression of $f$ by its interval counterpart. If each variable occurs only once in the expression of $f$ and $f$ is a composition of elementary functions, then the natural inclusion function is minimal [16]. In this article, we will use this *natural inclusion function* as the inclusion function for the nonlinear model.

### B. Contractor Programming

When the components of a vector $\mathbf{x}$ are linked by relations or constraints, a *Constraint Satisfaction Problem* (CSP) can be defined. It consists in finding the solution set $S = \{\mathbf{x} \in [\mathbf{x}] \mid g(\mathbf{x}) = 0\}$, where $[\mathbf{x}]$ is the domain of the variables and $g(\mathbf{x}) = 0$ represents the constraints. These constraints can also represent inequalities by introducing slack variables [8].

A *contractor* is a mapping from $\mathbb{IR}^n$ to $\mathbb{IR}^n$ such that

- $\forall [\mathbf{x}] \in \mathbb{IR}^n, C([\mathbf{x}]) \subseteq [\mathbf{x}]$ (**contraction**)
- $(\mathbf{x} \in [\mathbf{x}], C(\mathbf{x}) = \mathbf{x}) \Rightarrow x \in C([\mathbf{x}])$ (**consistency**)
- $C(\mathbf{x}) = \emptyset \Leftrightarrow (\exists \varepsilon > 0, \forall [\mathbf{x}] \subseteq B(\mathbf{x}, \varepsilon), C([\mathbf{x}]) = \emptyset)$ (**continuity**), where $B(\mathbf{x}, \varepsilon)$ is the ball centered on x with radius $\varepsilon$.

As illustrated in Fig. 2, contraction gives a sub-domain of the input domain $[\mathbf{x}]$. In this figure, the solution set $S$ of points satisfying the constraint is represented in red, the contraction of $[\mathbf{x}]$ by $C_1$ is represented in dashed.

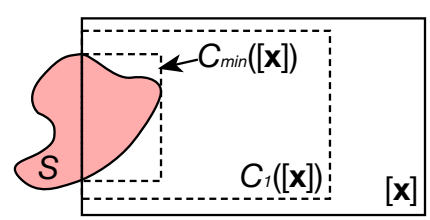


Fig. 2. Box before and after contraction

A contractor is *minimal* if the box resulting from the contraction is equal to the smallest interval that contains $S \cap C([\mathbf{x}])$. In Fig. 2, $C_{min}([\mathbf{x}])$ is the minimal contractor for $[\mathbf{x}]$ with respect to the constrain of belonging to $S$. From this example, we can see that different contractors can be defined for a CSP.

The most straightforward way to contract with respect to numerical constraints is by using the *forward-backward* algorithm. HC4 [18] is the classical *constraint propagation* loop found in the literature. It allows to contract with respect

to a system of constraints, by propagating constraints in an optimal order, as shown in the example of Fig. 3.

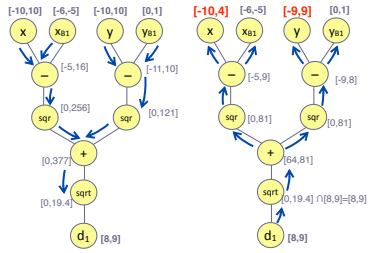


Fig. 3. Forward-Backward propagation with a distance constraint between two points: $d = \sqrt{(x - x_B)^2 + (y - y_B)^2}$

### C. Set Inversion Via Interval Analysis

Set inversion consists in computing the solution set $X = f^{-1}([\mathbf{y}]) = \{\mathbf{x} | f(\mathbf{x}) \in [\mathbf{y}]\}$, i.e the preimage of a box $[\mathbf{y}]$ by an arbitrary function $f$. The Set Inversion Via Interval Analysis (SIVIA) algorithm, introduced by Jaulin et al. in [8], employs a branch and bound strategy to approximate the solution set by a *subpaving* of an initial search domain. A subpaving is the union of non-overlapping and non-empty subboxes.

Computation of an outer approximation $\overline{X}$ of the solution set inside the initial domain $[\mathbf{x}_0]$ is presented in Alg. 1. The algorithm starts from an arbitrarily big initial box $[\mathbf{x}_0]$, and successively applies contractions and bisections to refine an outer approximation of solution set. SIVIA uses a working list of boxes $\mathfrak{L}$ generally implemented either as a stack for depth-first solution exploration, or as a queue for breadth-first computation. The contractor $\mathcal{C}_f$ is used to apply the constraint $f(\mathbf{x}) \in [\mathbf{y}]$ on each box. Each part of the box that has been discarded by contraction is guaranteed to be outside of the solution-set. The contractor may return an empty box if the current box is incompatible with the measurements. The sharpness of the outer subpaving $\overline{X}$ is controlled the parameter $\epsilon$. Boxes larger that $\epsilon$ after contraction are bisected and enqueued to be processed again. The choice of $\epsilon$ is a trade-off between the tightness of the solution-set approximation and the computation time.

**Algorithm 1** OuterSIVIA(in: $[\mathbf{x}_0]$, $\mathcal{C}_f$, $\epsilon$)

$\overline{X} := \emptyset$ *// empty subpaving*
put the box $[\mathbf{x}_0]$ into the list $\mathfrak{L}$
while $\mathfrak{L}$ is not empty
  $[\mathbf{x}] \leftarrow$ get the first box of $\mathfrak{L}$
  $[\mathbf{x}] \leftarrow \mathcal{C}_f([\mathbf{x}])$ *// contract the box*
  if $\text{width}([\mathbf{x}]) < \epsilon$ then
    $\overline{X} \leftarrow \overline{X} \cup [\mathbf{x}]$
  else if $x \neq \emptyset$ then
    $([\mathbf{x}_1], [\mathbf{x}_2]) \leftarrow \text{bisect}([\mathbf{x}])$
    put $[\mathbf{x}_1]$ in $\mathfrak{L}$; put $[\mathbf{x}_2]$ in $\mathfrak{L}$
  endif
end
return $\overline{X}$

## IV. Interval Based Pose Estimation

To solve the pose estimation problem addressed in Section II-A, instead of computing the pose by solving equation (3), our approach rely on seeking the domain of all the feasible poses $\mathbf{q}$ such that, $\mathbf{x}_i = \Pi\, {}^c\mathbf{T}_w(q)\, {}^w\mathbf{X}_i, i = 1..N$ according to uncertain 2D-3D point correspondences $\mathbf{x}_i$ and ${}^w\mathbf{X}_i$. The width of this domain will quantify the uncertainty on the pose with respect to that of the measurements. To achieve this goal, we define our problem as a Constraint Satisfaction Problem (CSP), and then use SIVIA with contractors to characterize the pose uncertainty domain.

### A. Pose Estimation from Image measurements

The projection transformation stated in Section II is a function of the robot's position in the world frame $\mathbf{q}$. In this article, we use Euler angle $ZYX(\phi, \theta, \psi)$ parameterization for the rotation matrix due to its minimal number of parameters to estimate (three) and also its suitability to represent a domain of angle uncertainty. With thus consider the pose vector $\mathbf{q} = (x, y, z, \phi, \theta, \psi)$, where $\phi, \theta, \psi$ are respectively the roll, pitch and yaw angles.

The projection equation (Eq. 2) and the use of a front looking camera give rise to 3 constraints associated to each pair of measurements; making the whole problem to have $3N$ constraints.

- The two constraints $C_{proj}$ expresses the correspondence between the image and world points. Recalling that our points are represented in homogeneous coordinates, and the image point coordinates in the normalized vector space, this constraints can be viewed as the set of robot states $\mathbf{q}$ that verifies the perspective projection relation. Let

$$({}^cX, {}^cY {}^cZ, 1)^T = {}^c\mathbf{T}_r\, {}^r\mathbf{T}_w(\mathbf{q})(X, Y, Z, 1)^T \quad (5)$$

  we have,

$$(u, v, 1)^T = \mathbf{K}\,\mathbf{\Pi}({}^cX, {}^cY {}^cZ, 1)^T \quad (6)$$

  where $({}^cX, {}^cY {}^cZ, 1)^T$ are the coordinates of the world point in the camera frame. $C_{proj}$ is expressed as follow:

$$C_{proj} = \{u = p_x x_i + u_0,\ v = p_y y_i + v_0\}$$

$$\text{with } x_i = \frac{{}^cX,}{{}^cZ,}, y_i = \frac{{}^cY,}{{}^cZ,}.$$

- The constraint $C_{front}$ expresses the fact that having a front-looking camera, all the points in the camera frame have the ${}^cZ$, coordinate positive (in other words, observed points are situated in the front half-space). This leads to:

$$C_{front} = \{{}^cZ, > 0\}$$

Let $C_i = \{C_{proj,i}, C_{front,i}\}$ be the set of all these constraints for the all the image-world pair $(\mathbf{x}_i, {}^w\mathbf{X}_i)$, with $i \in 1...N$, we can formulate the problem as a CSP $\mathcal{H}$ :

$$\mathcal{H} : \left( \ \mathbf{q} \in [\mathbf{q}], \mathbf{x}_i \in [\mathbf{x}_i], {}^w\mathbf{X}_i \in [{}^w\mathbf{X}_i], \{C_i,\ i \in 1...N\} \ \right), \quad (7)$$

where $[\mathbf{q}]$ represents prior knowledge of the pose, $[\mathbf{x}_i]$ represents bounded-error measurements of the landmarks in the image, and the width of $[{}^w\mathbf{X}_i]$ represents the uncertainty about world coordinates of the landmarks.

### B. Pose uncertainty domain computation

To compute the robot pose uncertainty domain, our algorithm takes the static 3D landmarks points present in the world frame which are visible by the camera, and their 2D image correspondents given by an image points features tracker. Considering bounded error measurements, each image point $\mathbf{x}_i$ and world point ${}^w\mathbf{X}_i$ can be represented as an interval vector, such that $\mathbf{x}_i \in [\mathbf{x}_i]$ and ${}^w\mathbf{X}_i \in [{}^w\mathbf{X}_i]$. The pose solution-set is computed as a subpaving, by using the *OuterSIVIA* set-inversion algorithm (Alg. 1). It employs a forward-backward contractor (HC4) built with the constraints of Eq. (7). The IBEX C++ library is employed for interval computations and contractors design.

The initial domain $[\mathbf{q}_0]$ of the pose before set-inversion is defined by prior available information. Since horizontal position is unknown, $[x_0]$ and $[y_0]$ are set to $[-\infty, +\infty]$. The altitude is given by a barometer altimeter and is therefore known to belong to an interval $[z_0]$ whose width corresponds to altitude uncertainty. Moreover, the embedded inertial measurement unit (IMU) provides reliable measurement of the roll and the pitch thanks to gyro and accelerometers, thus giving tight intervals for $[\phi_0]$ and $[\theta_0]$. Prior knowledge of the yaw is unavailable due to the lack of a compass, thus the initial domain $[\psi_0] = [-\pi, \pi]$.

### C. Prediction between images

As the robot is equipped with linear and rotational velocity sensors, the computed pose domain can be propagated from time to time by integrating the evolution equation. Assuming bounded measurement errors, measured velocities are represented by an interval vector $[\mathbf{v}] = ([v_x], [v_y], [v_z], [\omega_x], [\omega_y], [\omega_z])$. The upper (lower) bounds of the intervals are obtained by adding (subtracting) the sensor bounded error specification to the measured values.

Prediction then consists in integrating the following differential inclusion, where $\mathbf{f}$ denotes the robot evolution function

$$\dot{\mathbf{q}} \in \mathbf{f}(\mathbf{q}, [\mathbf{v}]).$$

Guaranteed integration techniques [19] can be employed to compute the domain of $[\mathbf{q}](t_{k+1})$ starting from the initial condition $\mathbf{q}(t_k) \in [\mathbf{q}](t_k)$.

Prediction from epoch to epoch of the pose uncertainty domain not only enables high-rate continuous positioning between camera acquisitions, but it also tightens the initial domain used for SIVIA, thus increasing computation speed. Figure 4 summarizes the flow of information used for pose domain computation at successive epochs.

### D. Fault detection and restart

A constraint satisfaction problem may have no solution. In the case of pose estimation, this happens when measurements are inconsistent, i.e when there is no feasible pose in the initial domain from which the current set of measurements could be obtained. The result of SIVIA is then an empty set. Such a situation arises either from modeling errors, underestimation of measurement error bounds, or the presence of spurious measurements. In real world situation, there is always a tradeoff when choosing error bounds, as it is not generally possible to fix a tight error bound that will cover even rare events.

In our implementation, if the set inversion with image measurements returns an empty pose domain, a "fault detected" flag is raised, and the next pose estimation will restart from the initial domain $[\mathbf{q}_0]$. Indeed, it is safer to fully restart estimation after fault detection to avoid propagating possible previously undetected faults through the prediction.

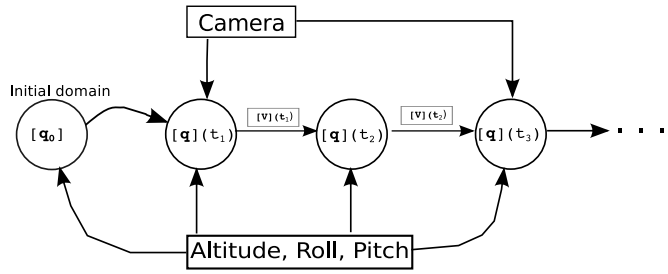


Fig. 4. Estimation and Prediction principle

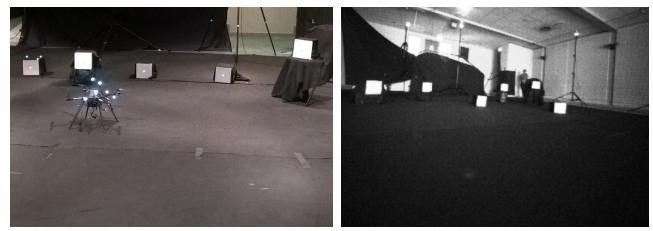

Fig. 5. *Left:* UAV in the experiment room. *Right:* Onboard camera view

## V. Experimental validation

This section presents the results of an experimental trial, conducted with a quadcopter UAV MK-Quadro from MikroKopter (left image of Fig. V-A). This quadcopter is equipped with an onboard camera for image acquisition.

### A. Experimental setup

Six cubes, whose position is known in the reference frame are set in the room, for use as landmarks. They present a 20 cm white square on a face for the purpose of image tracking. The center of the squares are considered for landmark coordinates. A Vicon motion capture system tracks the quadcopter's pose and the landmarks position.

A 77 seconds flight of the quadcopter has been done, including takeoff, a 3 m lateral movement and 50° rotation, returning home and landing, acquiring a total of 545 image frames. Square landmarks are tracked using a blob tracker, yielding rather noisy results. The tracking error, computed with respect to re-projected landmarks using the ground-truth pose from the motion capture system, is less than 4.5 px for 98% of the measurements. These errors are plotted in Fig. 6.

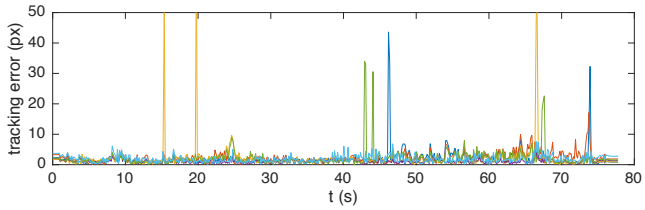


Fig. 6. Tracking error

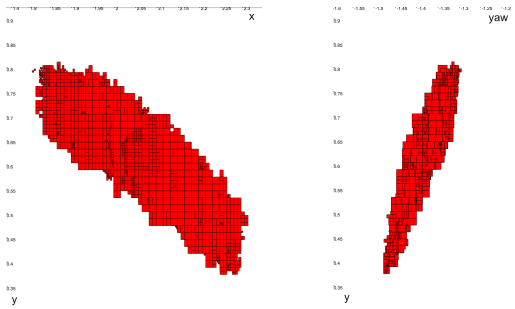


Fig. 7. View of the computed pose subpaving at t=0 s.
*Left:* projection on the $x, y$ plane. *Right:* projection on the $\psi, y$ plane.

There are some clearly noticeable spurious measurements of a few tenths of pixels during the try (see spikes in Fig. 6, e.g. at t=16 s and t=20 s). These correspond to failures of the landmark tracking algorithm. Smaller tracking errors of a few pixels are also present, they are mostly due to more complicated configurations when the landmarks are not easily distinguishable from the background.

### B. Pose domain computation

The pose domain computation is run over the whole trial. We set landmark tracking errors bounds to $\pm 4.5$ pixels in the image. This is consistent with 98% of the measurements as previously explained. Velocity measurement errors bounds are set to $\pm 0.1$ m/s linear and $\pm 0.005$ rad/s rotational. Landmarks positions are known within $\pm 1$ mm (Vicon precision).

The pose uncertainty domain subpaving computed from image measurements at time t=0 s is represented in Fig. 7. Figure shows $x, y$ plane and $\psi, y$ plane projections of this 6-dimensional domain. The $z$, $\phi$, $\theta$ components are not shown, since their bound are set as prior and are not modified during the set inversion. Indeed, this prior attitude and altitude information acts as a constraint on $x, y$ and $\psi$.

Figure 8 presents the results of the pose domain computation over the whole trial, for the $x, y$ and the $\psi$ components, and at image measurements epochs. The plotted values are relative to the ground truth, meaning that 0 is the ground truth. The blue lines are the upper and lower bounds of the domain. The sawtooth aspect of the yaw bounds (between t=30 s and t=40 s) corresponds to epochs when the set inversion process is not able to reduce the predicted yaw domain. It can be seen that computed uncertainty domain for $x$ lies within a radius of ±30 cm, $y$ within approximately ±20 cm and the yaw domain radius corresponds to ±5°.

The center dark line in Fig. 8 represents the midpoint of the computed pose domain. It can be used as a point estimate of the pose. Table I reports horizontal position and yaw error using this estimate. Results of an Extended Kalman Filter (EKF) on the same data are also shown for comparison. The point estimate of our method outperforms the EKF: this is in part due to the fault detection and restart implemented in our interval estimate.

TABLE I

COMPARISON OF OUR METHOD AND EKF: HORIZONTAL POSITION AND YAW ABSOLUTE ERRORS.

| Error | mean | median | 95% | max |
|---|---|---|---|---|
| Ours: yaw (deg) | 0.45 | 0.34 | 1.26 | 2.21 |
| Ours: 2D horizontal (cm) | 3.65 | 3.15 | 7.76 | 11.89 |
| EKF: yaw (deg) | 0.64 | 0.51 | 1.60 | 5.03 |
| EKF: 2D horizontal (cm) | 5.10 | 4.33 | 11.84 | 28.33 |

TABLE II

MEAN ERROR AND DOMAIN WIDTH W.R.T HEIGHT MEASUREMENT UNCERTAINTY

| height uncertainty | $\pm 1$ mm | $\pm 1$ cm | $\pm 5$ cm | $\pm 10$ cm |
|---|---|---|---|---|
| mean 2D error (cm) | 3.65 | 3.75 | 4.14 | 4.43 |
| mean yaw error (deg) | 0.45 | 0.45 | 0.46 | 0.50 |
| x domain width (cm) | 54.2 | 55.9 | 59.8 | 61.8 |
| y domain width (cm) | 40.8 | 43.0 | 47.6 | 49.8 |
| yaw domain width (deg) | 10.9 | 11.3 | 12.1 | 12.6 |

There are two important things to notice in these results. First, there some blanks in the the curves. They correspond to empty solution sets. An empty solution indicates inconsistency between the measurements, the error bounds and the model. These empty solutions occur when actual tracking error in the image is so large that the actual position of the landmark in the image is no more inside the domain $[\mathbf{x}_i]$. Detecting such inconsistencies enable fault detection. Fault detections are shown in Fig. 9. They clearly correspond to large tracking errors in Fig. 6. Over the whole trial, 2.6% of the epochs lead to fault detection (empty set).

The second remark concerns undetected faults. These missed detections occur when measurements errors are larger than the specified error bounds, but not enough to cause inconsistency with other measurements. This may lead to misleading information, as the computed uncertainty domain is no more guaranteed to contain the true robot pose in the missed detection case.

### C. Influence of altitude measurements uncertainty

There are various ways to measure altitude for an UAV. The most common are barometric sensors, ultrasonic range finders and laser telemeters. As those sensors give different uncertainties on the altitude measurement, we study in the sequel how altitude uncertainty affects the computed pose domain.

The proposed method has been tested with $\pm 1$ mm, $\pm 1$ cm, $\pm 5$ cm and $\pm 10$ cm altitude measurement intervals. The obtained mean pose errors and the mean domain width of the $x$, $y$ and $\psi$ components are shown in Table II.

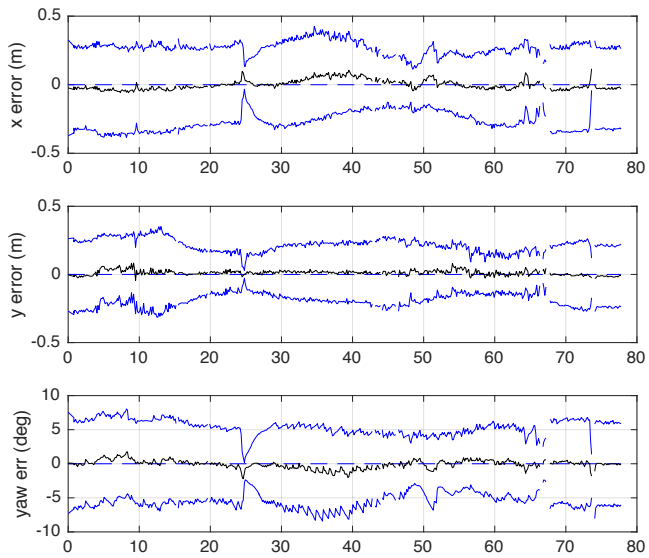


Fig. 8. *Blue:* error bounds of the $[x]$, $[y]$ and $[\psi]$ components of the computed pose domain. These are plotted relative to ground truth, i.e ground truth is 0.
*Black:* pose estimation error, using the center of the computed domain

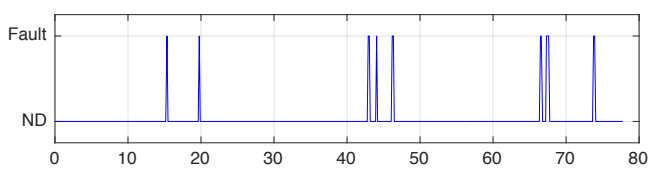


Fig. 9. Fault detection results

The results in Table II show that reducing uncertainty on the altitude measurements reduces the size of the computed pose domain. It can also be seen that not only pose uncertainty is reduced, but also the midpoint of the domain becomes a better point estimate. This clearly enlightens the benefit of a good onboard altitude sensor.

## VI. CONCLUSIONS

An interval based set-membership approach to pose estimation has been presented in this paper. It enables to compute a domain that contains the pose of an unmanned aerial vehicle (UAV), from uncertain bounded-error measurements of known landmarks in the image. Attitude and altitude information from onboard sensors are used as additional constraints on the pose uncertainty domain. Bounded error speed measurements enable prediction of the pose domain until the next image measurement epoch.

The proposed method relies on interval analysis and constraint propagation techniques to rigorously back-propagate the errors through the non-linear observation model. While interval methods provide guaranteed results as long as the measurement errors bounds are not violated, setting guaranteed measurement error bound in practice is generally impossible or very pessimistic. A fault detection system is thus implemented, in order to cope with inconsistencies due to spurious measurements (that do not respect the bounded error model).

Our method had been tested on a quadcopter flight, with noisy video tracking of landmarks. Fault detection enabled to handle tracking errors without returning a misleading pose domain. The center of the domain is a rather good point pose estimate, with a mean horizontal position error of less than 4 cm, and performs better than an EKF on the test data.

## REFERENCES


[1] M. Liu, C. Pradalier, Q. Chen, and R. Siegwart, "A bearing-only 2d/3d-homing method under a visual servoing framework," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 2010, pp. 4062–4067.

[2] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck, "Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle," *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 743–749, 2009.

[3] V. Lepetit and P. Fua, "Monocular model-based 3d tracking of rigid objects: A survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 1, no. 1, pp. 1–89, October 2005.

[4] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2001.

[5] E. Marchand, H. Uchiyama, and F. Spindler, "Pose estimation for augmented reality: a hands-on survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 12, pp. 2633–2651, 2016.

[6] V. Drevelle and P. Bonnifait, "Localization confidence domains via set inversion on short-term trajectory," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1244–1256, Oct 2013.

[7] B. Telle, M. J. Aldon, and N. Ramdani, "Guaranteed 3d visual sensing based on interval analysis," in *Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 2, Oct 2003, pp. 1566–1571 vol.2.

[8] L. Jaulin, M. Kieffer, O. Didrit, and E. Walter, *Applied interval analysis: with examples in parameter and state estimation, robust control and robotics*. Springer Verlag, 2001.

[9] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.

[10] N. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with application to image analysis and automated cartography," *Communication of the ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[11] D. Dementhon and L. Davis, "Model-based object pose in 25 lines of codes," *Int. J. of Computer Vision*, vol. 15, pp. 123–141, 1995.

[12] V. Lepetit, F. Moreno-Noguer, and P. Fua, "EPnP: An accurate O(n) solution to the PnP problem," *Int. Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[13] C. G. Bucher, "Adaptive sampling - an iterative fast monte carlo procedure," *Structural Safety*, vol. 5, no. 2, pp. 119 – 126, 1988.

[14] R. Melchers, "Importance sampling in structural systems," *Structural Safety*, vol. 6, no. 1, pp. 3 – 10, 1989. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0167473089900039

[15] S. H. Lee and W. Chen, "A comparative study of uncertainty propagation methods for black-box-type problems," *Structural and Multidisciplinary Optimization*, vol. 37, no. 3, p. 239, 2008.

[16] R. Moore, R. Kearfott, and M. Cloud, *Introduction to Interval Analysis*, ser. SIAM e-books. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2009. [Online]. Available: https://books.google.fr/books?id=tT7ykKbqfEwC

[17] V. Novák, I. Perfilieva, and J. Mockor, *Mathematical Principles of Fuzzy Logic*, ser. The Springer International Series in Engineering and Computer Science. Springer US, 2012. [Online]. Available: https://books.google.fr/books?id=uzPjBwAAQBAJ

[18] F. Benhamou, F. Goualard, L. Granvilliers, and J.-F. Puget, "Revising hull and box consistency," in *INT. CONF. ON LOGIC PROGRAMMING*. MIT press, 1999, pp. 230–244.

[19] J. A. dit Sandretto and A. Chapoutot, "Validated explicit and implicit runge-kutta methods," *Reliable Computing*, vol. 22, pp. 79–103, 2016.