



HAL
open science

Set-membership estimation and distributed control for a fleet of UAVs for target search and tracking

Maxime Zagar

► **To cite this version:**

Maxime Zagar. Set-membership estimation and distributed control for a fleet of UAVs for target search and tracking. Signal and Image processing. Université Paris-Saclay, 2024. English. NNT : 2024UPAST164 . tel-04831697

HAL Id: tel-04831697

<https://theses.hal.science/tel-04831697v1>

Submitted on 11 Dec 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Set-membership estimation and distributed control for a fleet of UAVs for target search and tracking

*Estimation ensembliste et commande distribuée de
flotte pour la recherche et le suivi de cibles*

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 580, sciences et technologies de l'information et de la
communication (STIC)
Spécialité de doctorat : Automatique
Graduate School : Sciences de l'ingénierie et des systèmes.
Référent : Faculté des sciences d'Orsay

Thèse préparée dans le **Laboratoire des signaux et systèmes** (Université Paris-Saclay,
CentraleSupélec, CNRS), sous la direction de **Michel KIEFFER** (Professeur), le
co-encadrement de **Hélène PIET-LAHANIER** (Directrice de recherche), et le
co-encadrement de **Luc MEYER** (Ingénieur de recherche).

Thèse soutenue à Paris-Saclay, le 4 Décembre 2024, par

Maxime ZAGAR

Composition du jury

Membres du jury avec voix délibérative

Philippe BONNIFAIT Professeur des universités, Université de Technologie de Compiègne	Président
Simon LACROIX Directeur de recherche, LAAS-CNRS/INSA Toulouse	Rapporteur & Examineur
Fabio MORBIDI Maître de conférences (HDR), Université de Picardie Jules Verne	Rapporteur & Examineur
Ouidad LABBANI-IGBIDA Professeure des universités, Université de Limoges	Examinatrice
Sylvie LE HEGARAT Professeure des universités, Université Paris-Saclay	Examinatrice

Titre : Estimation ensembliste et commande distribuée de flotte pour la recherche et le suivi de cibles

Mots clés : Estimation ensembliste, Estimation distribuée, Vision par ordinateurs, Commande prédictive, Flottes de drones

Résumé : Cette thèse aborde le problème de la recherche et du suivi d'un nombre inconnu de cibles mobiles réparties dans une région d'intérêt (Rdl) inconnue à l'aide d'une flotte coopérative de robots aériens sans pilote (UAVs). Ce problème de recherche, d'acquisition et de suivi coopératif (CSAT) apparaît dans des contextes militaires, e.g., pour trouver et suivre des véhicules ennemis, ou dans des applications civiles, e.g., pour rechercher des personnes perdues à la suite d'une catastrophe.

Pour résoudre ce problème, chaque drone embarque un système de vision par ordinateur (CVS) composé d'une caméra et d'algorithmes de traitement d'images qui fournissent des mesures liées à la Rdl. Les mesures fournies par le CVS consistent en des images avec des pixels labellisés, des cartes de profondeur et des boîtes contenant des pixels associés aux cibles qui ont été identifiées.

Cette thèse considère une approche ensembliste pour aborder le problème CSAT. Par rapport à d'autres approches reposant sur des hypothèses stochastiques sur le bruit de mesure, les approches ensemblistes supposent un bruit de mesure borné avec des bornes connues. Ces approches peuvent alors caractériser des ensembles qui contiennent nécessairement la position des cibles, à condition que les hypothèses sur les modèles de mesure et les bornes du bruit soient respectées. Très peu de travaux antérieurs exploitent directement les mesures CVS dans une approche ensembliste. Cela s'explique principalement par le fait qu'il est difficile d'obtenir des modèles de mesure pour un CVS lorsqu'il est composé d'algorithmes d'apprentissage profond. Pour résoudre ces problèmes, nous introduisons plusieurs hypothèses pour relier les mesures CVS aux cibles et aux obstacles présents dans la Rdl en utilisant une approche géométrique. À partir de ces hypothèses, nous proposons un nouvel estimateur ensembliste qui exploite di-

rectement les mesures CVS pour caractériser les ensembles qui sont garantis de contenir la position de chaque cible identifiée. Les mesures CVS sont également exploitées pour évaluer les ensembles qui sont garantis ne contenir aucune position de cible. Un processus de prédiction-correction similaire au filtre de Kalman a été implémenté pour tenir compte de l'échange d'informations entre les drones. La correction se fait via les mesures CVS acquises par chaque drone et des estimées transmises par les drones voisins. Plusieurs incertitudes supplémentaires peuvent être prises en compte dans l'approche proposée. Nous nous sommes concentrés sur la manière de prendre en compte l'incertitude de l'état des drones.

Comme la Rdl est encombrée d'obstacles inconnus, chaque drone établit une carte d'occupation et d'élévation pendant la recherche et le suivi des cibles à l'aide des mesures CVS. Cette carte fournit une description approximative de l'emplacement, de la hauteur, et de la forme des obstacles. La carte peut être exploitée pour éviter les obstacles. Dans cette thèse, elle est utilisée pour prédire la portion du champ de vue de la caméra de chaque drone qui est masquée par un obstacle. Cette information est déterminante dans la conception d'un algorithme par commande prédictive (MPC) pour déterminer la trajectoire de chaque drone, en minimisant l'incertitude de localisation des cibles identifiées et en réduisant la taille de l'ensemble contenant des cibles potentiellement non détectées. Si les cibles sont plus nombreuses que les UAVs, un compromis entre la recherche de nouvelles cibles et le suivi des cibles déjà identifiées afin de réduire l'incertitude de la localisation doit être trouvé.

Des simulations réalisées avec Webots illustrent les performances de l'estimateur et de la loi de guidage en évaluant l'efficacité des drones à trouver et identifier les cibles, et maintenir une estimation précise de leur position.

Title : Set-membership estimation and distributed control for a fleet of UAVs for target search and tracking

Keywords : Set-membership estimation, Distributed estimation, Computer vision, Model Predictive control, Fleet of UAVs

Abstract : This thesis addresses the problem of searching and tracking an unknown number of mobile targets spread within an unknown Region of Interest (RoI) using a fleet of cooperating Unmanned Aerial Vehicles (UAVs). Such Cooperative Search, Acquisition, and Tracking (CSAT) problem appears in military contexts, e.g., when enemy vehicles have to be found and tracked, or in civilian applications, e.g., when searching for lost people after a disaster.

To solve the CSAT problem, each UAV embeds a Computer Vision System (CVS) consisting of a camera and image processing algorithms that provide measurements related to the RoI. The CVS measurements consist of images with labeled pixels, depth maps, and boxes in the images containing pixels related to the detected and identified targets.

This thesis considers a set-membership approach to address the CSAT problem. Compared to alternative approaches relying on stochastic assumptions on the measurement noise, set-membership approaches assume bounded measurement noise with known bounds. Set-membership approaches can then characterize sets that are guaranteed to contain the location of the targets, provided that the hypotheses on the measurement models and noise bounds are satisfied. Very few previous works directly exploit CVS measurements in a set-membership approach. This is mainly because obtaining measurement models for a CVS involving deep learning algorithms is difficult. To address these issues, we introduce several assumptions to relate the CVS measurements with the targets and obstacles present in the RoI using a geometric approach. With these assumptions, we propose a new set-membership estimator that directly exploits the CVS measurements to characterize

sets that are guaranteed to contain the location of each identified target. The CVS measurements are also exploited to evaluate sets that are guaranteed to contain no target location. A prediction-correction scheme similar to the Kalman filter has been considered to account for the exchange of information between UAVs. The correction involves CVS measurements acquired by each UAV and estimates shared by neighboring UAVs. Several additional sources of uncertainty may be considered in the proposed approach. We have focused on the state uncertainty of UAVs.

As the RoI is cluttered with unknown obstacles, each UAV builds an occupancy-elevation map during the search and tracking of targets using CVS measurements. The map provides an approximate description of the location, height, and shape of the obstacles. The map may be exploited for obstacle avoidance. In this thesis, it is used to predict the occlusion by obstacles in the field of view of each UAV. This information is instrumental in the design of a Model Predictive Control (MPC) algorithm to determine the trajectory of each UAV, minimizing the localization uncertainty of identified targets and reducing the size of the set containing potentially undetected targets. If the targets outnumber the UAVs, a trade-off has to be found between searching for new targets and tracking those already identified to reduce the localization uncertainty.

Simulations performed with Webots illustrate the performance of the target location estimator and the MPC design by evaluating the efficiency of the cooperating UAVs to explore the environment, find and identify the targets, and maintain an accurate estimation of their location.

Résumé étendu

Cette thèse aborde le problème de la recherche et du suivi d'un nombre inconnu de cibles mobiles évoluant dans une Région d'Intérêt (RdI) encombrée d'obstacles de forme et de localisation inconnues. Une flotte de véhicules aériens sans pilote (UAV) est utilisée de manière coopérative et distribuée pour résoudre ce problème de recherche, d'acquisition et de suivi coopératifs (CSAT). Ce problème peut apparaître dans des contextes militaires, e.g., lorsqu'il s'agit de trouver et de suivre des véhicules ennemis, ou dans des applications civiles, e.g., lorsqu'il s'agit de rechercher des personnes perdues à la suite d'une catastrophe. L'une des difficultés de ce problème réside dans le fait que les cibles peuvent être plus nombreuses que les UAVs. Dans ce cas, il faut trouver un compromis entre l'exploration de la région d'intérêt pour rechercher de nouvelles cibles et la recapture des cibles identifiées afin de réduire l'incertitude de leur position estimée.

Le problème du CSAT est difficile à traiter car il combine plusieurs disciplines. Pour obtenir des mesures de l'environnement observé, chaque drone embarque un système de vision par ordinateur (CVS) composé d'une caméra et d'algorithmes de traitement d'images. Ces mesures contiennent des informations sur la présence ou l'absence d'une cible. Si une cible est détectée, un problème d'identification doit également être pris en compte. Ensuite, pour exploiter ces mesures CVS afin d'estimer les emplacements potentiels des cibles identifiées et non identifiées, un estimateur robuste doit être conçu. Enfin, chaque UAV résout le problème de compromis susmentionné pendant la mise à jour de la trajectoire en utilisant la théorie du contrôle optimal. Si elles sont disponibles, les communications avec les drones voisins sont exploitées à la fois dans le processus de localisation de la cible et durant la mise à jour de la trajectoire pour résoudre de manière coopérative le problème de la recherche et suivi de la cible. Toutes ces composantes du problème CSAT ont été abordées individuellement, mais peu de travaux les ont combinées.

Pour résoudre ce problème de CSAT, le CVS embarqué dans chaque drone consiste en une caméra et des algorithmes de traitement d'image qui fournissent des mesures liées à la RdI. Les mesures CVS consistent en des images avec des pixels étiquetés, des cartes de profondeur et des boîtes 2D localisant les cibles détectées et identifiées dans l'image. Un modèle de sténopé sans distorsion est utilisé pour modéliser la caméra et son champ de vision (FoV). À partir de cette approximation, une approche géométrique est proposée pour représenter les informations contenues dans une mesure CVS. Par exemple, si un pixel est étiqueté « sol », on peut supposer que ce pixel de la caméra embarquée n'a été éclairé que par des rayons lumineux provenant du sol. La formulation de ces hypothèses tire parti des limites implicites connues des mesures fournies, c'est-à-dire du fait qu'elles sont liées à un pixel de dimensions connues. Cela a motivé l'utilisation d'une approche ensembliste pour modéliser les bruits et autres incertitudes comme des ensembles bornés de bornes connues.

Cette thèse considère une approche ensembliste pour aborder le problème du CSAT. Par rapport à d'autres approches reposant sur des hypothèses stochastiques sur le bruit de mesure, les approches ensemblistes supposent un bruit de mesure borné avec des bornes connues. Ces approches peuvent alors caractériser des ensembles qui contiennent à coup sûr l'emplacement des cibles, à condition que les hypothèses sur les modèles de mesure et les bornes du bruit soient respectées. Très peu de travaux antérieurs exploitent directement les mesures CVS dans une approche ensembliste. Cela est principalement dû au fait qu'il est difficile d'obtenir des modèles de mesure pour un CVS reposant sur des méthodes d'apprentissage. Pour résoudre ces problèmes, nous introduisons plusieurs

hypothèses pour relier les mesures CVS aux cibles et aux obstacles présents dans le Rdl en utilisant une approche géométrique. Grâce à ces hypothèses, nous proposons un nouvel estimateur ensembliste qui exploite directement les mesures CVS pour caractériser les ensembles qui contiennent à coup sûr l'emplacement de chaque cible identifiée. Sous certaines hypothèses, certaines mesures CVS peuvent être exploitées en tant qu'informations négatives pour caractériser des ensembles dont on est sûr qu'ils ne contiennent aucune position de cible. Un processus de prédiction-corrrection similaire au filtre de Kalman a été utilisé pour tenir compte de l'échange d'informations entre les drones. La correction exploite les estimées ensemblistes caractérisés via les mesures CVS acquises par chaque drone ainsi que les estimées partagées par les drones voisins. En plus du bruit de mesure, nous proposons une approche pour rendre l'estimateur ensembliste robuste à l'incertitude de l'état des drones.

Comme la zone d'intérêt est encombrée d'obstacles inconnus, chaque drone exploite certaines mesures CVS pour construire durant la recherche et le suivi des cibles une carte d'occupation et d'élévation. Cette carte fournit une description approximative de l'emplacement, de la hauteur et de la forme des obstacles qui ont été détectés. La carte peut être exploitée pour éviter les obstacles. Dans cette thèse, elle est utilisée pour prédire la portion du sol que le drone ne pourra pas voir car cachée derrière un obstacle qu'il a déjà rencontré. Cette information est essentielle pour résoudre le problème de guidage afin de concevoir des trajectoires résolvant le compromis entre la recherche de cibles non identifiées et la recapture des cibles précédemment identifiées afin de réduire l'incertitude associée à leur position estimée. Pour résoudre ce problème, nous utilisons une loi de commande prédictive (MPC) pour déterminer la trajectoire de chaque UAV en minimisant l'incertitude de localisation des cibles identifiées et en réduisant la taille de l'ensemble contenant des cibles potentiellement non identifiées. Cette mise à jour de la trajectoire se fait de manière coopérative et distribuée.

Plusieurs simulations sont effectuées à l'aide de Webots pour simuler un environnement urbain. L'estimateur ensembliste et le MPC sont mis en œuvre sous Matlab. Les simulations illustrent les performances de l'estimateur et de la mise à jour des trajectoires en évaluant l'efficacité des drones coopératifs pour explorer l'environnement urbain inconnu, trouver et identifier les cibles, et maintenir une estimation précise de leur position.

Extended abstract

This thesis addresses the problem of searching and tracking of an unknown number of mobile targets spread within a Region of Interest (RoI) cluttered with obstacles of unknown shape and location. A fleet of cooperating Unmanned Aerial Vehicles (UAVs) is used in a cooperative and distributed manner to solve such Cooperative Search, Acquisition, and Tracking (CSAT) problem. This problem can appear in military contexts, e.g., when enemy vehicles have to be found and tracked, or in civilian applications, e.g., when searching for lost people after a disaster. A difficulty of this problem is the targets may outnumber the UAVs. In that case, a trade-off has to be solved between searching for new targets and recapturing identified targets to reduce the uncertainty of their estimated location.

The CSAT problem is difficult to address because it combines various disciplines. To get measurements from the perceived environment, each UAV embeds a Computer Vision System (CVS) consisting of a camera and image processing algorithms. These measurements contain information regarding the presence or absence of a target. Besides, if a target is detected, then an identification problem has to be also considered. Then, to exploit these CVS measurements in order to estimate the potential locations of identified and unidentified targets, a robust estimator needs to be designed. Finally, each UAV solves the aforementioned trade-off problem during the trajectory update by using optimal control theory. If available, communications with neighboring UAVs are exploited in both the target localization process and the trajectory update to solve cooperatively the target search and tracking problem. All of these components of the CSAT problem have been addressed individually, but few works have combined them all.

To solve this CSAT problem, the CVS embedding on each UAV consists of a camera and image processing algorithms that provide measurements related to the RoI. The CVS measurements consist of images with labeled pixels, depth maps, and bounding boxes localizing detected and identified targets in the image. A pinhole model without distortion is used to model the camera and its Field of View (FoV). From this approximation, a geometric approach is proposed to represent the information contained in a CVS measurement. For instance, if a pixel is labeled Ground, then one may assume that this pixel of the embedded camera has only been illuminated by light rays stemming from the ground. The formulation of these assumptions takes advantage of the known implicit bounds of the provided measurements, i.e., the fact that they are related to a pixel of known dimensions. This motivated the use of a set-membership approach to model noises and other uncertainties as bounded sets of known bounds.

This thesis considers a set-membership approach to address the CSAT problem. Compared to alternative approaches relying on stochastic assumptions on the measurement noise, set-membership approaches assume bounded measurement noise with known bounds. Set-membership approaches can then characterize sets that are guaranteed to contain the location of the targets, provided that the hypotheses on the measurement models and noise bounds are satisfied. Very few previous works directly exploit CVS measurements in a set-membership approach. This is mainly because obtaining measurement models for a CVS involving deep learning algorithms is difficult. To address these issues, we introduce several assumptions to relate the CVS measurements with the targets and obstacles present in the RoI using a geometric approach. With these assumptions, we propose a new set-membership estimator that directly exploits the CVS measurements to characterize sets that are guaranteed to contain the location of each identified target. Under some assumptions, some CVS

measurements can be exploited as negative information to characterize sets that are guaranteed to contain no target location. A prediction-correction process similar to the Kalman filter has been considered to account for the exchange of information between UAVs. The correction involves set estimates that are characterized using CVS measurements acquired by each UAV and estimates shared by neighboring UAVs. In addition of the measurement noise, we propose an approach to make the set-membership estimator robust to the state uncertainty of UAVs.

As the RoI is cluttered with unknown obstacles, each UAV builds an occupancy-elevation map during the search and tracking of targets using CVS measurements. The map provides an approximate description of the location, height, and shape of the obstacles. The map may be exploited for obstacle avoidance. In this thesis, it is used to predict the occlusion by obstacles in the FoV of the camera of each UAV. This information is instrumental to solve the guidance problem in order to design trajectories solving the trade-off between searching for unidentified targets and recapturing previously identified targets to reduce their estimated location uncertainty. To solve this problem, we design a Model Predictive Control (MPC) to determine the trajectory of each UAV minimizing the localization uncertainty of identified targets and reducing the size of the set containing potentially unidentified targets. This trajectory update is done in a cooperative and distributed manner.

Several simulations are preformed using Webots to simulate an urban environment. The set-membership estimator and the MPC are implemented on Matlab. The simulations illustrate the performance of the target location estimator and the MPC design by evaluating the efficiency of the cooperating UAVs to explore the unknown urban environment, find and identify the targets, and maintain an accurate estimation of their location.

Remerciements

Je tiens à remercier mon école, l'ENS Paris-Saclay, pour m'avoir formé, m'avoir donné goût à la recherche, et avoir financé cette thèse, me permettant ainsi de la réaliser dans les meilleures conditions possibles.

Je souhaite aussi remercier le L2S de CentraleSupélec ainsi que l'ONERA de Palaiseau pour avoir hébergé cette thèse et m'avoir permis de passer ces trois années de recherches dans des cadres exceptionnels.

Je tiens particulièrement à exprimer ma reconnaissance envers Monsieur Michel KIEFFER, mon directeur de thèse. Son expérience et son recul concernant la thématique de ma thèse, ainsi que des thématiques connexes, ont su efficacement m'orienter durant la résolution du problème multidisciplinaire qu'est la recherche et le suivi de cibles tout en me laissant une liberté d'approche plus qu'appréciable. De plus, son sens du détail fut un phare ayant éclairé ma plume lors de la rédaction de mes travaux. Compte tenu de sa disponibilité, du temps qu'il me consacra, et de l'attention qu'il me réserva durant ces 3 années, j'estime avoir été chanceux d'avoir eu un tel directeur de thèse. Il m'a réellement permis de prétendre à l'excellence

Je suis aussi particulièrement chanceux d'avoir eu Madame Hélène PIET-LAHANIER comme co-encadrante de cette thèse. Son expertise sur certains domaines la rendait complémentaire de Monsieur Michel KIEFFER. Son écoute et sa disponibilité furent aussi des atouts. Elle n'a pas ménagé son temps durant nos échanges ce qui m'a permis de clairement identifier là où mes idées péchaient et où elles devaient être renforcées. Grâce à ses contacts, j'ai eu aussi la chance de découvrir de nombreuses personnes à l'ONERA qui prirent aussi de leur temps pour échanger avec moi et discuter des idées que j'avais.

L'implication de Monsieur Luc MEYER comme second encadrant fut aussi déterminante durant ces 3 années. Naturellement, il mérite de partager plusieurs compliments que j'ai fait envers Monsieur KIEFFER et Madame PIET-LAHANIER. J'estime aussi que j'ai eu beaucoup de chance de l'avoir dans cette équipe encadrante pour le soutien moral qu'il m'apporta durant ces trois années.

Je souhaite aussi exprimer toute ma gratitude envers le personnel administratif tant du L2S que de l'école doctorale STIC qui furent toujours disponibles, même durant la période du piratage informatique de l'Université Paris-Saclay.

Concernant mes missions d'enseignements, je souhaite aussi remercier Monsieur Adrien MERCIER, Monsieur Mohammed Nabil EL KORSO, Monsieur Abdelhafid EL OUARDI, Monsieur Lucas GRECO, et Monsieur Christophe VIGNAT pour m'avoir accueilli dans l'équipe pédagogique et avec qui j'ai passé trois années enrichissantes auprès des élèves durant l'encadrement de leurs TP et projets.

Il va sans dire que je remercie également mes parents, ma petite sœur, et Mélanie, pour tout le soutien qu'ils m'ont apporté durant ces trois ans.

Table des matières

1	Introduction	25
1.1	Context	25
1.2	Thesis outline and contributions	28
1.3	Publications	30
2	State of the art	31
2.1	Target search and tracking using mobile robots	31
2.1.1	Mission classification	32
2.1.2	Cooperative Search, Acquisition, and Track	33
2.1.3	Problems related to the target search and tracking problem	34
2.2	Sensors and computer vision algorithms	35
2.2.1	Examples of sensors	36
2.2.2	Examples of computer vision algorithms	37
2.2.3	Accuracy of computer vision algorithms	38
2.3	Map building	40
2.4	Target localization	44
2.4.1	Deterministic approach	44
2.4.2	Stochastic approach	45
2.4.3	Set-membership approach	46
2.4.4	Negative information	47
2.4.5	Idealized CVS	48
2.5	Target search and tracking	49
2.5.1	Representation of the target state	49
2.5.2	Target search	50
2.5.3	Target pursuit	53
2.5.4	Target tracking	54
2.6	Control design	56
2.6.1	Coordination between the robots	56
2.6.2	Examples of control design approaches	57
2.6.3	Predictive control	59
2.7	Conclusion of the chapter	62
3	Hypotheses and Problem formulation	63
3.1	Environment model	63
3.2	Target model	65
3.2.1	Interaction of a target with its environment	65
3.3	UAV model	67
3.3.1	Communication	68

3.4	Model of the camera and CVS	69
3.4.1	Camera model	69
3.4.2	Field of view	70
3.4.3	Projection on camera frame	71
3.5	Information provided by the CVS	72
3.5.1	Depth map	72
3.5.2	Pixel classifier	73
3.5.3	Target detection and identification	77
3.6	Assumption on observed targets	77
3.7	Problem formulation	78
3.8	Conclusion	79
4	Target localization from CVS measurements	81
4.1	Problem formulation	81
4.2	Exploiting CVS information to localize an identified target	82
4.3	Exploiting CVS information to characterize a target-free area	85
4.3.1	Using pixels labeled Ground	85
4.3.2	Using pixels labeled Obstacle	86
4.3.3	Using pixels labeled as Target	88
4.4	Estimation of the hidden area	89
4.5	Estimation algorithm	90
4.5.1	Prediction step	90
4.5.2	Update using CVS measurements	90
4.5.3	Update after communication with neighbors	92
4.6	Update of the occupancy-elevation map	92
4.6.1	Definition of an occupancy-elevation map	93
4.6.2	Update using CVS information	94
4.6.3	Update after communication with neighbors	95
4.6.4	Approximation of $\mathbb{H}_{i,k}^{CVS}$ using the OEM	95
4.7	Simulation	96
4.7.1	Simulation conditions and metrics	97
4.7.2	Location estimation of a fully-visible target	98
4.7.3	Exploiting the minimum distance between targets	103
4.7.4	Target localization in the presence of obstacles	106
4.7.5	Inner approximation of the r_S^{to} -ground neighborhood of an obstacle	107
4.7.6	OEM construction	108
4.8	Conclusion	111
5	Trajectory design for target search and tracking	113
5.1	Problem formulation	114
5.2	Trajectory design for cooperative exploration	115
5.2.1	Trajectory design ignoring the presence of obstacles	115
5.2.2	Considering obstacles	120

5.2.3	Exploiting the OEM	122
5.2.4	Promoting the exploration of recently hidden areas	123
5.3	MPC implementation	124
5.3.1	Practical choices	124
5.3.2	Task scheduling	126
5.4	Simulation	128
5.4.1	Simulation parameters	129
5.4.2	Evaluation metrics	130
5.4.3	Target search and tracking performance	132
5.4.4	Exploration efficiency	135
5.4.5	Computing time	138
5.5	Conclusion	139
6	Target localization with UAV state uncertainties	141
6.1	Problem formulation	141
6.1.1	UAV state uncertainty	141
6.1.2	Exploiting CVS measurements	142
6.2	Interval analysis	144
6.2.1	Inclusion functions and set inversion	145
6.2.2	Thick intervals	146
6.3	Thick function of \mathbf{h}^{CVS}	148
6.3.1	Upper bound for the maximum of a norm	150
6.3.2	Upper bound for the maximum of a ratio	150
6.4	Characterization of $\mathbb{P}_i((n_r, n_c))$ and $\mathbb{P}_i^g((n_r, n_c))$ accounting for UAV state uncertainties	152
6.4.1	Characterization of $\mathbb{P}_i((n_r, n_c))$	152
6.4.2	Characterization of $\mathbb{P}_i^g((n_r, n_c))$	152
6.5	Inverse of the measurement function	153
6.5.1	Bijection	153
6.5.2	Exploitation of the bijection	154
6.6	Simulation results	154
6.6.1	Simulation conditions	154
6.6.2	Adaption of the set-membership estimator	155
6.6.3	Characterization of a set estimate containing no target location	156
6.6.4	Fully-visible Target location estimation	157
6.6.5	Inner approximation of the r_s^{to} -ground neighborhood of an obstacle	161
6.7	Conclusion	162
7	Conclusion	165
8	Perspectives	169
8.1	Imperfect target detection and identification	169
8.1.1	Proposed approach	169
8.2	Uneven ground	170

8.2.1	Proposed approach	171
8.3	Relaxation of the \mathcal{Z} -convex obstacle assumption	171
8.3.1	Proposed approach	171

A Appendix 173

A.1	Alternative target location estimator	173
A.1.1	Target localization - Assumption 2	173
A.1.2	Target localization - Assumption 3	173
A.1.3	Exploiting CVS information to characterize a target-free area	174
A.1.4	Simulation	175
A.2	Practical implementation	175
A.2.1	Characterization of $\mathbb{P}_{i,k}((n_r, n_c))$	176
A.2.2	Characterization of $\mathbb{X}_{i,j,k}^{t,m}$	176
A.2.3	Characterization of $\mathbb{S}((n_r, n_c), r_s^{to})$	177
A.2.4	Characterization of $\mathbb{X}_{i,k}^o$ and $\mathbb{X}_{i,j,k}^{t,m}$	177
A.2.5	Characterization of $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$ and $\mathbb{H}_{i,k}^{CVS}$	178
A.3	Interval Analysis and SIVIA	178
A.3.1	Preliminaries	178
A.3.2	Inclusion function	179
A.3.3	Set Inverter Via Interval Analysis	180
A.4	Thick intervals & Thick-SIVIA	182
A.4.1	Set inversion reformulation	182
A.4.2	Thick sets and thick inclusion function	182
A.4.3	Thick SIVIA without \mathbb{X}^\vee	186
A.4.4	Thick SIVIA with \mathbb{X}^\vee	186

Table des figures

2.1	Illustration of the chain to get new knowledge from an observed scene. A UAV is monitoring an area with cars and obstacles. Its embedded CVS provides new measurements such as an image, bounding boxes (purple) of the moving objects, a segmentation, or a depth map. Then, these CVS measurements are used to estimate the location of the cars (green sets) and the location of the obstacles (blue sets).	36
2.2	Snapshots of the exploration mission presented in (Williams et al., 2020). The magenta dots are candidate locations that the robot may choose to reach to continue the exploration. The white line is the trajectory that the robot took. The red line shows the path to the selected candidate location. The point cloud represents the portion of the indoor environment that has been mapped.	37
2.3	Illustration of the detection results obtained with the YOLOD algorithm developed in (Luo et al., 2022). Red boxes are for detected cars. Purples boxes are for detected pickups. Orange boxes are for detected trucks. Cyan boxes are for detected boats. Green boxes are for detected camping cars.	38
2.4	Illustration from (Liu et al., 2022a) of the detection of human targets using YOLOv3 and the identification of the correct target to track with the Bi-directional Similarity Matching (BSM) algorithm (Oron et al., 2017).	38
2.5	Illustration of the IoU metric. A car is visible the image. The blue box is the ground truth, and the orange box is what the target detector (trained to detect cars) returns. The IoU is the ratio between the intersection between the two boxes and the area of their union.	39
2.6	Illustration of the performance of a 6D-pose algorithm. Red boxes are for cars and green boxes are for cyclists. Most of the estimated locations, orientations, and dimensions are correct. Nonetheless, in the top-right figure, the algorithm provided an erroneous pose estimation of the partially-visible black car. Figure taken from (Mousavian et al., 2017).	40
2.7	Figure from (Asgharivaskasi and Atanasov, 2023) illustrating the grid-based representation (via semantic Octomap) of the RoI during its exploration.	41
2.8	Figure from (Zhang et al., 2022) illustrating the maps that are considered during the exploration and mapping of the environment using a UAV. With the elevation map, other properties of the terrain are evaluated : the slope, the roughness, and the step height. These metrics are evaluated using (Stelzer et al., 2012). Once obtained, the traversability map is deduced.	42
2.9	Figure from (Zhang et al., 2023). Illustration of the assumption related to the pyramidal stacking of parallelepipedic blocks to model urban buildings.	43
2.10	Figure from (Zhang et al., 2023). Illustration of the building reconstruction approach with a point cloud initially obtained from CVS measurement (in blue) and the fictive point cloud generated from the pyramidal assumption (in red).	43

2.11	Figure from (Souza and Goncalves, 2016). Illustration of the sensor model. A beam is cast in a direction. The cells of the OEM that are flown over are detected, and a probability of being occupied by an obstacle is assigned to them. This occupancy probability accounts for a measured distance.	44
2.12	Figure from (Di Gennaro and Waldmann, 2023) where three detected targets evolving within the overlapping FoV of static cameras have to be localized using stereoscopy techniques. Target 1 has been detected. However, their algorithm has poorly estimated its location.	45
2.13	Empirical study of YOLO Tiny 2.0 to detect the presence of a human-like target. The chosen metric to evaluate the accuracy of the target detector is the F1 score. Figure taken from (Meera et al., 2019).	46
2.14	Illustration of a 2D TPM which has been taken from the simulation results of (Li et al., 2024).	50
2.15	Illustration from (Ibenthal et al., 2020) of the evolution at 4 different time instants of the possible locations of targets (black dots) that have been identified (green sets) and not-identified (yellow sets) by a fleet of 4 UAVs. The white circles are decoys that have not to be localized. The gray polygons represent the obstacles.	51
2.16	Illustration of the searching strategy presented in (Niedzielski et al., 2021) where 4 RoIs (not represented) have been defined and explored using a lawnmower trajectory. The orange star and the blue pentagon are two locations that have been used to define the areas where the lost man may be. The red dot is the location where he has been found by the UAV.	52
2.17	Illustration from (Kuhlman et al., 2017) of an uneven ground generated by an elevation grid	53
2.18	Illustration taken from (Ibenthal et al., 2023) which represents the area theoretically covered by the 4 UAVs (light gray). A point on the ground is guaranteed to be observed only if it has been observed simultaneously by 4 UAVs where each one is located within one of the conic observation subsets represented by the geometrical observation conditions illustrated by the subset \mathbb{C}_1 (green), \mathbb{C}_2 (red), \mathbb{C}_3 (blue), and \mathbb{C}_4 (purple). The dark gray set is the area that has been proved to be observed even in the presence of unknown obstacles.	53
2.19	Figure from (Li et al., 2021). The nonlinear transformation between the probability that a target exists in a given cell, <i>i.e.</i> , $\text{TPM}_k(r, c)$, and what they defined as the uncertainty of the cell, <i>i.e.</i> , $\mu_k(r, c) = \exp(-k \ln(1/\text{TPM}_k(r, c)) - 1)$	58
3.1	Illustration of different obstacles that may or may not satisfy equation (3.1).	64
3.2	Illustration of the urban ROI simulated on Webots (Webots,) used to evaluate the performance of our approach in Chapter 5. Only buildings satisfying (3.1) are considered.	64
3.3	Illustration of a possible vertical circular right cylinder $\mathbb{C}^t(\mathbf{x}_{j,k}^{tg})$ containing the shape $\mathbb{S}^t(\mathbf{x}_{j,k}^t)$ of a car-like or truck-like target	66
3.4	Illustration of the r_s^{to} -ground neighborhood (in red) of two obstacles (houses, in gray). Target is in green and its location is the black dot.	67

3.5	Illustration of (3.11). The red sets are the boundaries of the r_s^{tt} -ground neighborhoods of the two target locations (black dots).	68
3.6	Illustration of the pinhole model : $c = -f_c x_1^{\mathcal{F}^c} / x_3^{\mathcal{F}^c} + N_c/2$ and $r = -f_r x_2^{\mathcal{F}^c} / x_3^{\mathcal{F}^c} + N_r/2$.	70
3.7	Some light rays illuminate of the pixel (n_r, n_c) ; The 4 blue lines are the light rays illuminating the corners of (n_r, n_c) ; The quadrangle $\mathbb{P}_i^g((n_r, n_c))$ is defined in Section 4.3.1. .	71
3.8	According to Assumption 4, if pixel $\mathbf{p}_1 \in \mathcal{Y}_i^o$, then all light rays illuminating \mathbf{p}_1 are assumed to stem from an obstacle. If pixel $\mathbf{p}_3 \in \mathcal{Y}_i^g$, then all light rays illuminating \mathbf{p}_3 are assumed to stem from \mathbb{X}_g . If $\mathbf{p}_2 \in \mathcal{Y}_i^n$, then all light rays illuminating \mathbf{p}_2 are assumed to stem from a different entities. The orange lines represent the light rays \mathbf{v} used to obtain the noise-free distance \mathbf{D}_i^0 (orange dots) for each of the 3 pixels. The blue dots correspond to $\mathbf{D}_i(\mathbf{p}_\ell)$ and the black intervals are $[\mathbf{D}_i](\mathbf{p}_\ell)$, with $\ell \in \{1, 2, 3\}$, see (3.35).	75
3.9	According to Assumption 5, if pixel $\mathbf{p}_1 \in \mathcal{Y}_i^o$, then at least one light ray illuminating \mathbf{p}_1 stems from an obstacle. If pixel $\mathbf{p}_3 \in \mathcal{Y}_i^g$, then at least one light ray illuminating \mathbf{p}_3 stems from \mathbb{X}_g . If $\mathbf{p}_2 \in \mathcal{Y}_i^n$, nothing can be said about light rays illuminating \mathbf{p}_2 . If $\mathbf{p}_2 \in \mathcal{Y}_i^o$ or \mathcal{Y}_i^g , then at least one light ray illuminating \mathbf{p}_2 is assumed to stem from an obstacle or the ground respectively. Assuming $\mathbf{p}_2 \in \mathcal{Y}_i^o$, then the purple set represents $\mathcal{Y}_i^o(\mathbf{p}_2)$. The orange lines represent the light rays \mathbf{v} used to obtain the noise-free distance \mathbf{D}_i^0 (orange dots) for each of the 3 pixels. The blue dots correspond to $\mathbf{D}_i(\mathbf{p}_\ell)$ and the black intervals are $[\mathbf{D}_i](\mathbf{p}_\ell)$, with $\ell \in \{1, 2, 3\}$, see (3.35).	76
3.10	According to Assumption 6, if pixel $\mathbf{p}_1 \in \mathcal{Y}_i^o$, then the light ray illuminating the center of \mathbf{p}_1 stems from an obstacle. If pixel $\mathbf{p}_3 \in \mathcal{Y}_i^g$, then the light ray illuminating the center of \mathbf{p}_3 stems from \mathbb{X}_g . If $\mathbf{p}_2 \in \mathcal{Y}_i^o$, then the light ray illuminating the center of \mathbf{p}_2 stems from an obstacle. The red lines represent the light rays $\mathbf{v}^c(\mathbf{p}_\ell)$ illuminating the center of \mathbf{p}_ℓ that has been used to obtain $\mathbf{D}_i^0(\mathbf{p}_\ell)$ (red dots), with $\ell \in \{1, 2, 3\}$. The blue dots correspond to $\mathbf{D}_i(\mathbf{p}_\ell)$ and the black intervals are $[\mathbf{D}_i](\mathbf{p}_\ell)$, see (3.35).	76
3.11	Illustration of (3.54). The blue dot represents \mathbf{x}_j^t and the green dot represents \mathbf{x}_j^{tg} . The dashed-red line is the line-of-sight $[\mathbf{x}_i^c, \mathbf{x}_j^{\text{tg}}]$. The purple dot represents the part of $\mathbb{S}^t(\mathbf{x}_j^t)$ seen by the UAV.	78
4.1	Illustration of the sets $\mathbb{P}_i((n_r, n_c))$ (in orange) for different $(n_r, n_c) \in \mathcal{Y}_{i,j}^t \cap [\mathcal{Y}_{i,j}^t]$ and for a target detected and identified by UAV i	83
4.2	Estimation of the location of target j : The set $\mathbb{P}_{i,j}^t$ is projected on the ground (red set). Then $\mathbf{p}_g(\mathbb{C}^t)$ (horizontal circles represented by the black lines) is used to build $\mathbb{X}_{i,j}^{\text{t,m}}$ (green set). The green dot represents \mathbf{x}_j^t	85
4.3	Illustration of the set $\mathbb{P}_i^g(\mathcal{Y}_i^g)$ (in orange) defined as a set which cannot contain the location (in green) \mathbf{x}_j^{tg} , for $j \in \mathcal{N}^t$, of any target	86
4.4	Illustration of Proposition 16 : The obstacle \mathbb{S}_m^o and the projection on the ground $\mathbf{p}_g(\mathbb{S}_m^o)$ of its shape \mathbb{S}_m^o are represented by the light-green sets. The set $\mathbb{P}_i((n_r, n_c))$ associated to a pixel (n_r, n_c) labeled <i>Obstacle</i> and its ground projection are represented by the orange sets. $\mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{to}})$ is represented by the black segments. The red set is $\underline{\mathbb{S}}((n_r, n_c), r_s^{\text{to}})$ and the dark-green set is $\mathbb{N}_g(\mathbf{p}_g(\mathbb{S}_m^o), r_s^{\text{to}})$	88

- 4.5 Representation of an occupancy-elevation map $\mathcal{M}_{i,k}$. Green cells represent *empty* cells; the portions of \mathbb{X}_g covered by those cells are considered free of obstacles. Blue cells are *unexplored* cells; no useful information related to the map topology is available. The red parallelepipeds are obtained from the location and size of the *occupied* cells (an obstacle has been detected on those cells) and the measured height (related to those detected obstacles). 93
- 4.6 Illustration of the update of the OEM. The blue pixels are labeled *Obstacle*. From those pixels, UAV i characterizes their $\mathbb{P}_{i,k}(\cdot)$ (in orange). The blue and purple dots are their \bar{x} and h_{\max} . The green pixel is labeled *Ground*. The green set is the $\mathbb{P}_{i,k}^g(\cdot)$ related to this pixel and the green dot is the center $c(\eta)$ where the cell η is the only cell that has its center contained in $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$. Assuming that each cell was previously unexplored, the fifth and sixth cells are updated as *occupied* with the correct elevation. The fourth cell is updated as *empty*. 95
- 4.7 Illustration of the portion of the ground occluded by an obstacle approximated by the red box $\mathbb{B}_{i,k}^M(\eta)$. The UAV location is represented by the blue star, the set $\mathbb{F}(\mathbf{x}_{i,k}^u) \cap \mathbb{X}_g$ is delimited in black, and the set $\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_{i,k}^u, \eta)$ is delimited in purple. Thus, the portion of $\mathbb{F}(\mathbf{x}_{i,k}^u)$ that is occluded by $\mathbb{B}_{i,k}^M(\eta)$ (the set delimited in green) is the intersection between $\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_{i,k}^u, \eta)$ and $\mathbb{F}(\mathbf{x}_{i,k}^u)$. Figure taken from (Reboul et al., 2019). 96
- 4.8 CVS information provided to a UAV by the sensors simulated by Webots. On the left : the image \mathbf{I}_i with two bounding boxes. At the center : the array of pixel labels \mathbf{L}_i with two bounding boxes (*Target*-labeled pixels in green, *Obstacle*-labeled pixels in blue, *Ground*-labeled pixels in brown, and *Unknown*-labeled pixels in black). On the right : the depth map \mathbf{D}_i 98
- 4.9 Illustration of some of the CVS measurements that a UAV can acquire in a situation with no obstacles. The UAV location and attitude is $(0, 0, 50, 0, 0, 0)^\top$. The location of the targets is : $(60, 0, 0)^\top$, $(90, 0, 0)^\top$, $(120, 0, 0)^\top$, $(150, 0, 0)^\top$, and $(180, 0, 0)^\top$ 99
- 4.10 Representation of the set estimates $\mathbb{P}_i^g(\mathcal{Y}_i^g)$ (in blue), $\mathbf{p}_g(\mathbb{P}_{i,j}^t)$ (in red), and $\mathbb{X}_{i,j}^{\text{t,m}}$ (in green) obtained after the processing of the CVS measurements with the set-membership estimator. The black dots are the different target locations. 99
- 4.11 Mean value and standard deviation over 100 simulations of $\phi^{\text{eR}}(\mathbb{X}_{i,j}^{\text{t,m}})$ (in green), $\phi^{\text{eR}}(\mathbb{X}_{i,j}^{\text{t}})$ (in black), and $\phi^{\text{c}}(\mathbb{X}_{i,j}^{\text{t}})$ (in red). 101
- 4.12 Representation of the set estimates $\mathbb{P}_i^g(\mathcal{Y}_i^g)$ (in blue), $\mathbb{X}_{i,j}^{\text{t,m}}$ (in green), and $\mathbb{X}_{i,j}^{\text{t}}$ (in yellow). The black dots represent the different target locations. The red dots correspond to $c(\mathbb{X}_{i,j}^{\text{t}})$ and the circle of center $c(\mathbb{X}_{i,j}^{\text{t}})$ and radius $\phi^{\text{eR}}(\mathbb{X}_{i,j}^{\text{t}})$ is in red. 101
- 4.13 Mean value with standard deviation over 100 simulations of $\phi^{\text{eR}}(\mathbb{X}_{i,j}^{\text{t,m}})$ (in green), $\phi^{\text{eR}}(\mathbb{X}_{i,j}^{\text{t}})$ (in black), and $\phi^{\text{c}}(\mathbb{X}_{i,j}^{\text{t}})$ (in red). The left subfigure is for $r^{\text{t}} = 2.5$ m. The middle subfigure is for $r^{\text{t}} = 3$ m. The right subfigure is for $r^{\text{t}} = 4$ m. 102

- 4.14 Mean value with standard deviation over 100 simulations of $\phi^{\text{eR}} \left(\mathbb{X}_{i,j}^{\text{t,m}} \right)$ (in green), $\phi^{\text{eR}} \left(\mathbb{X}_{i,j}^{\text{t}} \right)$ (in black), and $\phi^{\text{c}} \left(\mathbb{X}_{i,j}^{\text{t}} \right)$ (in red). The left subfigure is for $[\underline{w}, \bar{w}] = [-0.01, 0.01]$. The middle subfigure is for $[\underline{w}, \bar{w}] = [-0.05, 0.05]$. The right subfigure is for $[\underline{w}, \bar{w}] = [-0.1, 0.1]$. 103
- 4.15 Illustration of CVS measurements with a camera of resolution $N_r \times N_c = 720 \times 960$ pixels 104
- 4.16 Representation of the set estimates $\mathbb{P}_i^{\text{g}} \left(\mathcal{Y}_i^{\text{g}} \right)$ (in blue), $\mathbf{p}_g \left(\mathbb{P}_{i,j}^{\text{t}} \right)$ (in red), and $\mathbb{X}_{i,j}^{\text{t,m}}$ (in green) that are obtained from the measurements provided by the embedded CVS when the camera provides images of resolution $N_r \times N_c = 720 \times 960$ pixels. The black dots are the different target locations. 104
- 4.17 Mean value and standard deviation over 100 simulations of $\phi^{\text{eR}} \left(\mathbb{X}_{i,j}^{\text{t,m}} \right)$ (dashed lines - curves at the top of the figure), $\phi^{\text{eR}} \left(\mathbb{X}_{i,j}^{\text{t}} \right)$ (solid lines - curves in the middle of the figure), and $\phi^{\text{c}} \left(\mathbb{X}_{i,j}^{\text{t}} \right)$ (broken lines - curves at the bottom of the figure). The green curves represent the results obtained with a camera of resolution $N_r \times N_c = 720 \times 960$ pixels; the blue curves represent the results obtained with a camera of resolution $N_r \times N_c = 360 \times 480$ pixels. 105
- 4.18 Representation of the set estimates $\mathbb{P}_{i,k}^{\text{g}} \left(\mathcal{Y}_{i,k}^{\text{g}} \right)$ (in blue), $\bar{\mathbb{X}}_{i,k}^{\text{t}} \cap \mathbb{P}_{i,k}^{\text{g}} \left(\mathcal{Y}_{i,j,k}^{\text{t}} \right)$ (in yellow), and $\mathbb{X}_{i,j,k}^{\text{t,m}}$ (in green). The black dots are the different target locations. Here, $r_s^{\text{tt}} = 6 \text{ m}$. 105
- 4.19 Mean value with standard deviation over 100 simulations of $\Phi^{\Delta} \left(\mathbb{X}_{i,j,k}^{\text{t,m}}, \bar{\mathbb{X}}_{i,k}^{\text{t}} \cap \mathbb{P}_{i,k}^{\text{g}} \left(\mathcal{Y}_{i,j,k}^{\text{t}} \right) \right)$ for different values of r_s^{tt} 106
- 4.20 Illustration of the CVS measurements when a target is located close to an obstacle (in the image). 107
- 4.21 Estimation of the location of a target which seems to be close to an obstacle (c.f. Figure 4.20). The red dot represents the UAV location. $\bar{\mathbb{X}}_{i,k}^{\text{t}}$ is in yellow, $\mathbb{X}_{i,j,k}^{\text{t}}$ is in green, and $\mathbb{X}_{i,j,k}^{\text{t,m}}$ is represented by the set with the pink border. The set $\mathbb{X}_{i,k}^{\text{o}}$ representing an inner approximation of r_s^{to} -ground neighborhood of an obstacle is in gray. 107
- 4.22 Illustration of the CVS measurements when numerous obstacles are located within the FoV of the UAV. 108
- 4.23 Illustration of \mathbb{X}_i^{o} obtained for different values of r_s^{to} . The red dot represents the UAV location and the set with the black border is $\mathbb{F} \left(\mathbf{x}_i^{\text{u}} \right) \cap \mathbb{X}_g$ 109
- 4.24 Example of a scene simulated on Webots where several buildings and houses are located within the RoI 109
- 4.25 CVS measurements available to the UAV when it observes the scene in Figure 4.24. The UAV state (location and attitude) is $(0, 0, 50, 0, 0, 0)^{\text{T}}$ 109
- 4.26 Representation of an OEM after processing the CVS measurements reported in Figure 4.25. The blue squares represent the *Unexplored* cells. The green squares represent the *Empty* cells. The red parallelepipeds result from the *Occupied* cells and the registered height. The UAV location is represented by the black dot with a pink outline. 110

4.27	Illustration of $\mathbb{H}_i^{\text{OEM}}(\mathbf{x}_i^u) \cap \mathbb{F}(\mathbf{x}_i^u)$ (in red) and $\mathbb{H}_i^{\text{CVS}}$ (in blue) when the OEM is built with only the CVS measurements acquired at time t_k . In the left subfigure, the black boxes represent the set of boxes (4.41), and the red set is $\mathbb{H}_i^{\text{OEM}}(\mathbf{x}_i^u)$, where \mathbf{x}_i^u is the black dot. In the right subfigure, $\mathbb{H}_i^{\text{CVS}}$ overlaid $\mathbb{H}_i^{\text{OEM}}(\mathbf{x}_i^u) \cap \mathbb{F}(\mathbf{x}_i^u)$	111
5.1	Illustration of the design of a trajectory to explore the set $\overline{\mathbb{X}}_{i,k}^t$ (in yellow) when the UAV (in red) is not aware of the presence of an obstacle (in gray). The blue UAVs represent three possible next UAV locations.	116
5.2	Impact of the simplification $\mathbb{H}_{i,k+\tau k}^{\text{CVS}} = \mathbb{H}_{i,k}^{\text{CVS}}$ (red hatched set), for $\tau = 1, \dots, h$ in the MPC trajectory design : the UAV location at time t_k is in red, the candidate trajectories and final locations are in blue or purple. Trajectories such that the FoV intersects $\mathbb{H}_{i,k}^{\text{CVS}}$ (left part) will be avoided due to the reduction of the usable part of the FoV in favor of trajectories without intersection between the FoV and $\mathbb{H}_{i,k}^{\text{CVS}}$ (right part). The purple hatched sets are the $\mathbb{H}_{i,k+h}^{\text{CVS}}$ obtained at each location represented by the blue UAVs.	121
5.3	Evolution of the yaw angle of a UAV in response to a yaw angle step of $\pi/3$ radian (or 60 degrees).	125
5.4	Illustration of two trajectory updates, one with $N^{\text{MPC}} = 2$ (left), the other with $N^{\text{MPC}} = 1$ (right), while having $h' = 3$. The purple circle is the current location of UAV i (and also $\hat{\mathbf{x}}_{i,k k-h'}^u$). The blue and black circles are the locations of UAV i according to the sequence of control inputs obtained at time $t_{k-h'}$ whereas the red circles are the locations of UAV i according to the sequence obtained at time t_k . The difference between the blue and black circles are that the black circles are actually representing the trajectory that UAV i has taken, whereas the blue circles are what remains of the previous trajectory, <i>i.e.</i> , the fragment of $\hat{\mathbf{x}}_{i,k-h':h k-h'}^u$ before the trajectory update at time t_k . For the sake of simplicity, the targets are not moving here.	126
5.5	Comparison between an approach where the prediction of the set estimates is done with a period T (right) and (left) an approach where the set estimates are only updated with a period $T^{\text{MPC}} = 3T$. The light green and light yellow part of the set estimates represent the observed parts. Targets are assumed static.	127
5.6	Succession of tasks performed by three UAVs; here the trajectory update with the MPC is performed sequentially.	127
5.7	Succession of tasks performed by three UAVs; here the trajectory update with the MPC is performed using only the previously known trajectories of the UAV's neighbors	128
5.8	Simulated urban environment in Webots	129
5.9	(Left) Time evolution of $\text{card}(\mathcal{L}_{i,k}^t)$ for each considered criterion. (Right) Evolution of the number of targets located in the FoV of at least one UAV (black), and of the number of identified targets, <i>i.e.</i> , $\text{card}(\bigcup_{\ell \in \mathcal{N}^u} \mathcal{D}_{\ell,k}^t)$, (green). Only criterion (5.16) (top-right) (5.32) (bottom-right) are considered. These curves are average of 10 simulations.	133
5.10	Mean value over 10 simulations for each criterion of $e_{i,k}^t$ (left), $\Phi_{i,k}^t$ (top-right) and $\Phi_{i,k}^{t,eR}$ (bottom-right). To lighten the figure, the standard deviation is not shown.	134
5.11	Mean value over 10 simulations for each criterion of $\phi(\overline{\mathbb{X}}_{i,k}^t) / \phi(\mathbb{X}_g)$ (solid lines), $\phi(\overline{\mathbb{X}}_{i,k}^t \cap \mathbb{X}_{i,k}^h) / \phi(\mathbb{X}_g)$ (dashed lines). To lighten the figure, the standard deviation is not shown.	135

5.12	Mean value of $\Phi_k^{\text{FoV}}/\phi(\mathbb{X}_g)$ (black) and of $\Phi_k^g/\phi(\mathbb{X}_g)$ (green) when only criteria (5.16) (top) and (5.32) (bottom) are considered	136
5.13	Mean value over 10 simulations for each criterion of $\Phi_{i,k}^{\text{CG}}/\phi(\mathbb{X}_g)$. To lighten the figure, the standard deviation is not shown.	137
5.14	Mean value and standard deviation over 10 simulations of Φ_k^{OEM} when only criterion (5.32) is considered	138
5.15	Mean value over 10 simulations of the time required for a UAV to run the set estimator from CVS measurements (left) and the MPC approach (right)	139
6.1	Illustration of a thick box $\llbracket \mathbf{x} \rrbracket = \llbracket [\mathbf{x}^-], [\mathbf{x}^+] \rrbracket$ (not represented) and a box $[\mathbf{x}]$ (the dashed box). The boxes $[\mathbf{b}_1]$, $[\mathbf{b}_2]$, and $[\mathbf{b}_3]$ are included in $[\mathbf{x}]$, but $[\mathbf{b}_3]$ is not included in $\llbracket \mathbf{x} \rrbracket$ because its lower and upper bounds (represented by dots) are not in $[\mathbf{x}^-]$ and $[\mathbf{x}^+]$. . .	147
6.2	Illustration of some of the CVS measurements that a UAV can acquire in a situation with no obstacles. All targets are identified and the pink boxes represent their location in the image.	155
6.3	Representation of $\left(\mathbb{P}_{i,k}^g(\mathcal{N}^l)\right)^\exists$ where \mathcal{B}_{in} is in green, $\mathcal{B}_{\text{penumbra}}$ is in blue, and $\mathcal{B}_{\text{border}}$ is in yellow. The red set is the true intersection of the UAV FoV with the ground when the state is perfectly known.	157
6.4	Illustration of $\left(\mathbb{P}_i^g(\mathcal{N}^l)\right)^\forall$ for different values of UAV state uncertainties. The set with a black border is $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$. The black dots represent the 5 target locations.	157
6.5	Comparison between two different approaches to get an outer-approximation of $\left(\mathbb{X}_{i,j}^{\text{t,m}}\right)^\exists$. The set with a red border is obtained by an approach that does not use Thick SIVIA whereas the blue set is obtained with an approach that uses Thick SIVIA. The green set represents the set $\mathbb{X}_{i,j}^{\text{t,m}}$ that would be obtained if there are no UAV state uncertainties. The black dot represents the target location.	158
6.6	Comparison of the set estimates that would be obtained without UAV state uncertainties (first line), with the set estimates that are obtained while considering an uncertain UAV state vector (second line). For both lines, $\mathbb{P}_i^g(\mathcal{Y}_i^g)$ and $\left(\mathbb{P}_i^g(\mathcal{Y}_i^g)\right)^\forall$ are represented by the blue sets, $\mathbf{p}_g\left(\mathbb{P}_{i,j}^{\text{t}}\right)$ and $\left(\mathbf{p}_g\left(\mathbb{P}_{i,j}^{\text{t}}\right)\right)^\exists$ are represented with by the sets with a red border, and $\mathbb{X}_{i,j}^{\text{t,m}}$ and $\left(\mathbb{X}_{i,j}^{\text{t,m}}\right)^\exists$ are represented with the sets with a green border. The black dots represent the target locations.	159
6.7	On the left : Mean value and standard deviation over 100 simulations of $\phi^{\text{eR}}\left(\mathbb{X}_{i,j}^{\text{t,m}}\right)$ (in green), $\phi^{\text{eR}}\left(\mathbb{X}_{i,j}^{\text{t}}\right)$ (in black), and $\phi^{\text{c}}\left(\mathbb{X}_{i,j}^{\text{t}}\right)$ (in red). The UAV state vector is assumed perfectly known. On the right : Mean value with standard deviation over 10 simulations of $\phi^{\text{eR}}\left(\left(\mathbb{X}_{i,j}^{\text{t,m}}\right)^\exists\right)$ (in green), $\phi^{\text{eR}}\left(\left(\mathbb{X}_{i,j}^{\text{t}}\right)^\exists\right)$ (in black), and $\phi^{\text{c}}\left(\left(\mathbb{X}_{i,j}^{\text{t}}\right)^\exists\right)$ (in red). Uncertainty 1 is considered to model the UAV state uncertainties.	160
6.8	Representation of the set estimates $\left(\mathbb{P}_i^g(\mathcal{Y}_i^g)\right)^\forall$ (in blue), $\left(\mathbb{X}_i^{\text{t}}\right)^\exists \cap \left(\mathbb{P}_i^g(\mathcal{Y}_{i,j}^{\text{t}})\right)^\exists$ (in yellow), and $\left(\mathbb{X}_{i,j}^{\text{t,m}}\right)^\exists$ (in green). The black dots are the different target locations. Here, $r_s^{\text{tt}} = 8 \text{ m}$	161

6.9	Comparison of $(\mathbb{X}_i^0)^\exists$ for different values of r_s^{to} and UAV state uncertainties. The red dot represents the actual location of the UAV and the set with a black border represents $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$	162
8.1	Estimation of the location of a target (black dot) from several set estimates. The green sets are the set estimates that are guaranteed to contain the target location. The orange set estimate represents an outlier. The left part shows a situation where no outlier is present. The middle part shows a situation with one outlier. However, since the intersection of the 3 sets is not empty, the error cannot be detected. The right part shows a similar situation but where the intersection of the 3 sets is empty. The presence of the outlier can be detected and rejected using q -relaxed intersection.	170
8.2	Illustration of the city setup with uneven ground created on Webots for the simulation in Chapter 5.	171
8.3	Illustration of a T-shape obstacle observed by a UAV. The green area is $\{\mathbf{x} \in \mathbb{X}_0 \mid x_3 \in [0, h^t]\}$. Pixels \mathbf{p}_1 and \mathbf{p}_2 are labeled <i>Obstacle</i> . \mathbf{p}_1 cannot be related to the base of the obstacle because $\mathbb{P}_{i,k}(\mathbf{p}_1)$ (purple set) does not intersect with $\mathbb{S}_m^{\text{o,base}}$. \mathbf{p}_2 can be related to the base of an obstacle because $\mathbb{P}_{i,k}(\mathbf{p}_2)$ (orange set) is included in the green area.	172
A.1	Illustration of an issue with $\mathbb{P}_i((n_r, n_c))$ (in orange) for a <i>target</i> -labeled pixel $(n_r, n_c) \in \mathcal{Y}_i^t \cap [\mathcal{Y}_{i,j}^t]$ while considering Assumption 5 but without assuming that the vector \mathbf{v} such that $\mathbf{D}_i^0(n_r, n_c) = \rho(\mathbf{x}_i^c, \mathbf{v})$ belongs to $\mathcal{V}_i^t(n_r, n_c)$ (the purple set). Without that assumption, even though the pixel is labeled <i>Target</i> , $\mathbb{P}_i((n_r, n_c))$ does not intersect with the shape of a target.	174
A.2	Illustration of the sets $\mathbb{P}_i((n_r, n_c))$ (in orange) for different <i>target</i> -labeled pixels $(n_r, n_c) \in \mathcal{Y}_i^t \cap [\mathcal{Y}_{i,j}^t]$ while considering Assumption 6.	174
A.3	Mean value with standard deviation over 100 simulations of $\phi^{\text{eR}}(\mathbb{X}_{i,j,k}^{\text{t,m}})$ (dashed lines - curves at the top of the figure), $\phi^{\text{eR}}(\mathbb{X}_{i,j,k}^{\text{t}})$ (solid lines - curves in the middle of the figure), and $\phi^c(\mathbb{X}_{i,j,k}^{\text{t}})$ (broken lines - curves at the bottom of the figure). The green curves are for Assumption 6. The blue curves are for Assumption 4 and 5.	175
A.4	Illustration of \mathbb{X}^\forall and \mathbb{X}^\exists . The black sets are potential solution sets for a specific value of the vector of parameters.	179
A.5	Illustration of an inclusion function and the pessimism induced by this approximation.	180
A.6	Characterization of the set defined by (A.21) using SIVIA. \mathcal{B}_{in} is the 304 green boxes. \mathcal{B}_{out} is the red boxes. $\mathcal{B}_{\text{border}}$ is the 920 yellow boxes.	181
A.7	Characterization of the set defined by (A.21) using SIVIA while considering an uncertain vector of parameters. \mathcal{B}_{in} is the 236 green boxes. \mathcal{B}_{out} is the red boxes. $\mathcal{B}_{\text{border}}$ is the 8672 yellow boxes.	181
A.8	Illustration a thick box $\llbracket \mathbf{h}_i \rrbracket([\mathbf{x}]) = \llbracket [\mathbf{h}_i^-]([\mathbf{x}]), [\mathbf{h}_i^+]([\mathbf{x}]) \rrbracket$ and boxes such that $[\mathbf{b}_1] \in \llbracket \mathbf{h}_i \rrbracket([\mathbf{x}])$ and $[\mathbf{b}_2] \in \llbracket \mathbf{h}_i \rrbracket([\mathbf{x}])$ but $[\mathbf{b}_3] \notin \llbracket \mathbf{h}_i \rrbracket([\mathbf{x}])$	184

A.9	Characterization of the set defined by (A.21) using Thick SIVIA while considering an uncertain vector of parameters. $\mathcal{B}_{\text{penumbra}}$ is the 272 blue boxes. \mathcal{B}_{out} is the red boxes. $\mathcal{B}_{\text{border}}$ is the 912 yellow boxes.	186
A.10	Characterization of the set defined by (A.21) using Thick SIVIA while considering an uncertain vector of parameters. \mathcal{B}_{in} is the 236 green boxes. $\mathcal{B}_{\text{penumbra}}$ is the 1752 blue boxes. \mathcal{B}_{out} is the red boxes. $\mathcal{B}_{\text{border}}$ is the 1752 yellow boxes.	187

Notations

Notation	Signification
Rol	Region of Interest
OEM	Occupancy-Elevation Map
UAV	Unmanned Aerial Vehicle
FoV	Field of View
CVS	Computer Vision System
CSAT	Cooperative Search, Acquisition, and Tracking
MPC	Model Predictive Control
SIVIA	Set Inverter Via Interval Analysis
TIF	Thick Inclusion Function
i index	related to UAV i
j index	related to target j
m index	related to obstacle m
k index	related to time t_k
u superscript	related to a UAV
t superscript	related to a target
o superscript	related to an obstacle

Notation	Definition
$\mathbf{p}_g(\cdot)$	Projection on \mathbb{X}_g
$\mathbf{p}_{\mathcal{F}_i^c}(\mathbf{x}^{\mathcal{F}_i^c})$	Projection on the CCD array of a point $\mathbf{x}^{\mathcal{F}_i^c} \in \mathbb{R}^3$, see (3.17)
$\mathbb{N}(\mathbb{S}, r)$	r -neighborhood within the Rol of a set \mathbb{S} , see (3.5)
$\mathbb{N}_g(\mathbb{S}, r)$	r -ground-neighborhood within the Rol of a set $\mathbb{S} \subset \mathbb{X}_g$, see (3.6)
$\mathbf{f}^t(\cdot) / \mathbf{f}^u(\cdot)$	dynamics of a target / a UAV
$\mathbf{M}_{\mathcal{F}_1}^{\mathcal{F}_2}$	Rotation matrix from \mathcal{F}_1 to \mathcal{F}_2
$\mathbf{T}_{\mathcal{F}_1}^{\mathcal{F}_2}(\mathbf{x})$	Coordinates of some vector $\mathbf{x} \in \mathbb{R}^3$, initially expressed in \mathcal{F}_1 , expressed in \mathcal{F}_2
$\mathbf{p}_c(\mathbf{x}_{i,k}^u, \mathbf{x})$	2D coordinates of a point \mathbf{x} when projected onto the CCD array of UAV i , see (3.27)
$\rho(\mathbf{x}_{i,k}^c, \mathbf{v})$	Distance between $\mathbf{x}_{i,k}^c$ and $\mathbb{S}_k^{\text{Rol}}$ along a $\mathbf{v} \in \mathcal{V}_{i,k}(n_r, n_c)$, see (3.31)
$\mathbb{P}_{i,k}((n_r, n_c))$	Set estimate containing all points that may have illuminated pixel (n_r, n_c) of the camera of UAV i while being at a distance consistent with $[\mathbf{D}_{i,k}](n_r, n_c)$, see (4.1)
$\mathbb{P}_{i,k}^g((n_r, n_c))$	Set estimate containing all points of \mathbb{X}_g that may have illuminated pixel (n_r, n_c) of the camera of UAV i , see (4.5)

Notation	Definition
$\mathcal{F}/\mathcal{F}_i^b/\mathcal{F}_i^c$	Reference frame / Frame attached to the body / the camera of UAV i
$\mathcal{N}^u/\mathcal{N}^t/\mathcal{N}^o$	List of the indexes of the N^u UAVs / N^t targets / N^o obstacles
\mathcal{N}^l	List of the coordinates of each pixel of the camera
\mathcal{N}^M	List of the indexes of each cell composing the OEM
$\mathcal{N}_{i,k}$	List of the indexes of the UAVs that can communicate with UAV i at time t_k
$\mathbf{x}_{j,k}^t/\mathbf{x}_{i,k}^u$	State vector of target j / UAV i at time t_k
$\mathbf{x}_{j,k}^t$	Center of gravity of target j at time t_k
$\mathbf{x}_{j,k}^{t,g}$	Projection on the ground of $\mathbf{x}_{j,k}^t$
$\mathbf{x}_{i,k}^u/\Theta_{i,k}^u$	Center of gravity / attitude of UAV i at time t_k
$\mathbf{x}_{i,k}^c$	Optical center of the camera of UAV i at time t_k
$\mathbb{S}^u(\mathbf{x}_{i,k}^u)/\mathbb{S}^t(\mathbf{x}_{j,k}^t)/\mathbb{S}^o_m$	Shape of the i -th UAV/ j -th target / m -th obstacle at time t_k
$\mathbb{C}^t(\mathbf{x}_{j,k}^{t,g})/h^t/r^t$	Circular right cylinder of height h^t and radius r^t approximating $\mathbb{S}^t(\mathbf{x}_{j,k}^t)$
$\mathbb{X}_0/\mathbb{X}_g$	Rol / Ground of the Rol
$\mathbb{S}_k^{\text{Rol}}$	Set containing all points belonging to either \mathbb{X}_g , a target, a UAV, or an obstacle
$\mathcal{M}_{i,k}$	OEM known to UAV i at time t_k
$r_s^{\text{to}}/r_s^{\text{tt}}$	Safety distance between ($\{\text{to}\}$) a target and an obstacle, ($\{\text{tt}\}$) two targets
$\mathcal{V}_{i,k}(n_r, n_c)$	Set of light rays $\mathbf{v}^{\mathcal{F}_i^c}$ illuminating pixel (n_r, n_c)
$\mathbb{F}(\mathbf{x}_{i,k}^u)/\underline{\mathbb{F}}(\mathbf{x}_{i,k}^u)$	(unlimited / bounded) FoV of UAV i at time t_k
$\mathbf{I}_{i,k}/\mathbf{L}_{i,k}/\mathbf{D}_{i,k}$	Image, labeled image, and depth map provided by the CVS of UAV i at time t_k
$\mathbf{D}_{i,k}^0(n_r, n_c)$	Noise free depth measurement associated to pixel (n_r, n_c)
$[\mathbf{D}_{i,k}]_i(n_r, n_c)$	Interval containing $\mathbf{D}_{i,k}^0(n_r, n_c)$
$\mathcal{Y}_{i,k}^g/\mathcal{Y}_{i,k}^t/\mathcal{Y}_{i,k}^o/\mathcal{Y}_{i,k}^n$	List of indexes of the pixels labeled <i>Ground</i> / <i>Target</i> / <i>Obstacle</i> / <i>Unknown</i> .
$\mathcal{Y}_{i,j,k}^t$	Bounding box localizing target j within $\mathbf{I}_{i,k}$
$\mathcal{D}_{i,k}^t$	List of target indexes identified by UAV i at time t_k
$\mathcal{L}_{i,k}^t$	List known to UAV i at time t_k of the indexes of identified and localized targets
$\mathbb{X}_{i,j,k}^t$	Set estimate known to UAV i at time t_k containing the location of target j
$\mathcal{X}_{i,k}^t$	List containing $\mathbb{X}_{i,j,k}^t$ for all $j \in \mathcal{L}_{i,k}^t$
$\overline{\mathbb{X}}_{i,k}^t$	Set estimate known to UAV i at time t_k containing the location of possible not yet identified targets.
$\mathbb{X}_{i,k}^o$	Set estimate known by UAV i at time t_k that cannot contain $\mathbf{x}_{j,k}^{t,g}$ for $j \in \mathcal{N}^t$ due to the presence of an obstacle
$\mathbb{X}_{i,j,k}^{\text{t,m}}$	Set estimate characterized from the CVS measurements available to UAV i at time t_k that contains $\mathbf{x}_{j,k}^{t,g}$ for $j \in \mathcal{D}_{i,k}^t$
$\mathbb{P}_{i,k}(\mathcal{Y}_{i,k}^g)$	Set estimate characterized from the CVS measurements available to UAV i at time t_k that cannot contain any target location
$\overline{\mathbb{X}}_{i,k}^o$	Set estimate representing a region around the obstacles detected by UAV i at time t_k that cannot contain any target location
$\overline{\mathbb{X}}_{i,j,k}^{\text{t,m}}$	Set estimate representing a region around target j that cannot contain any target location except $\mathbf{x}_{j,k}^{t,g}$ with $j \in \mathcal{D}_{i,k}^t$
$\mathbb{H}_{i,k}^{\text{CVS}}$	Set estimate representing the part of \mathbb{X}_g that UAV i has not observed at time t_k
$\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_{i,k}^u)$	Set estimate that approximates $\mathbb{H}_{i,k}^{\text{CVS}}$ by using the OEM

Chapter 1 - Introduction

1.1 . Context

This thesis addresses the problem of target Search, Acquisition, and Tracking (SAT), which has been formalized 70 years ago in (Koopman, 1956). Initially developed by the United States Navy to search, detect, identify, and localize hostile vehicles for their termination, the introduced *theory of search* has led to the foundations of a more civilian problem where one or several targets evolving in a Region of Interest (RoI) have to be found and tracked. Nevertheless, and regardless of the application, SAT problems have always overlapped with other problems such as efficient positioning in the environment, exploitation of noisy measurements for target detection, target identification, implementation of algorithms for accurate and robust target localization, deployment of strategies involving exploration of a hazardous environment, target pursuit, and energy management. Each aspect has been addressed and developed. For instance, the efficient positioning problem, *i.e.*, finding the best location to either efficiently cover an area or to get new and useful information regarding its exploration, has been addressed 40 years ago with an Art-Gallery problem (O'Rourke and others, 1987). Nowadays, other approaches have been proposed such as the Next-Best-View approach (Batinovic et al., 2022) that combines a shadow-casting approach, initially used in video games, with a cubic approximation of the map to search for the best path to take to explore and map an environment. The target detection and its identification have been formalized in (Johnson, 1985). From that, several methods have been proposed to address a target detection problem such as (Girshick et al., 2014; Liu et al., 2016; Redmon et al., 2016), and recently, deep-learning algorithms have been developed for target detection, and pose estimation (Fan et al., 2022). This tendency to continuously develop new methods, or algorithms, to solve the SAT problem has been motivated by several factors. Drones, their embedded sensors, and the algorithms to process the information collected by these sensors become more sophisticated. Moreover, the available computing power and storage capacity continue to improve.

Additionally, put aside the SAT problem, other problems related to searching or pursuing a target exist and can be either addressed with these new algorithms and technologies, or be the foundation of new methods. Consider as an example a simpler target tracking problem which involves a robot that is tasked to keep the target within its sensing range. To address this problem, a new algorithm for target detection and identification has been proposed and tested (Liu et al., 2022a). Target hunting problems, as in (Vidal et al., 2002; Olsen et al., 2022), assume that the targets are evasive and thus evolve within the environment in order to avoid detection. To solve this problem, more specific strategies based on, *e.g.*, game theory (Basar and Olsder, 1998) have been used.

Observation of multiple moving targets using multiple robots is a 30-years-ago problem introduced in (Parker and Emmons, 1997) that is close to a SAT problem. Both problems involve the search of targets in an unknown environment, but once a target is identified, a robot stops the target search and is assigned to only pursue the newly discovered target. A SAT problem is different because it involves two conflicting objectives : the target search and the target pursuing, with a limited number of robots. On the one hand, target search requires an efficient exploration of the environment in order to detect

and identify new targets, this task becoming even more difficult if the number of targets to identify is unknown. On the other hand, target pursuing requires the requisition of a robot to recapture a previously identified target in order to reduce its estimated location uncertainty. This is because, since the dynamic of mobile targets is often at least partly unknown, the estimated location of an identified target becomes more uncertain when not monitored. Consequently, a trade-off between exploration and pursuit has to be addressed in order to accomplish the SAT mission. In this context, (Furukawa et al., 2006) have introduced and formalized the Cooperative Search, Acquisition, and Tracking (CSAT) problem and proposed a first solution algorithm. The CSAT problem requires that each robot in the fleet takes an action that contributes to solve the initial SAT problem while being beneficial to the whole fleet, rather than of acting alone. Coordination between the robots is therefore essential.

This thesis proposes new methods and strategies to solve a CSAT problem. Unfortunately, even though CSAT algorithms have numerous applications, where some of them are life-saving during Search And Rescue (SAR) missions, there still exist few real-life applications. An example of SAR mission that has been achieved with the help of a UAV has been described in (Niedzielski et al., 2021). A family has reported to have lost contact with one of their elders in a Polish town. Though the rescue team deployed an Unmanned Aerial Vehicle (UAV) to assist the terrestrial search by exploring identified areas that may contain the lost man, the exploration followed a lawnmower trajectory, the UAV took photos of the area, and their processing was done offline once uploaded on a laptop. Moreover, a human operator has been assigned to correct the false detection provided by their algorithm. The rescue has been successful but has required almost 4 hours. This work - which claims to be the first successful application of a target detector in a real search and rescue mission - highlights the inherent high complexity of CSAT missions and the number of subproblems that have to be considered perform target search in an automatic way. Some of these subproblems are reliable image processing, robust estimation, UAV control design, and navigation within an unknown cluttered environment. Several approaches addressing one of these subproblems already exist, but they focus on a specific aspect of the CSAT problem, such as target location estimation from an image or efficient exploration of an unknown environment, and thus do not consider the whole CSAT problem. On the other hand, strategies to solve a CSAT problem have been developed, but most of them consider simplifying assumptions regarding target detection and location estimation.

The target localization problem includes several sub-problems. One of those is collecting new information from the environment that is under observation, *e.g.*, an urban area with cars. Then, processing this collected information to acquire new measurements, *e.g.*, an image or a depth map. Lastly, transforming these measurements through an estimation process to gain new knowledge regarding, *e.g.*, the possible location of the cars, before the update of previous knowledge up to the current time. Each of those sub-processes has been already addressed and tested in real-life while providing clear limits for their performances. For instance, as pointed out in (Symington et al., 2010; Meera et al., 2019), the performance in target detection is dependent on the observation conditions (including the presence of obstacles) and weather conditions. Nonetheless, weather conditions are likely to be under-represented in training data sets for deep-learning algorithms such as (Redmon et al., 2016). Additionally, aerial images often involve small targets with a complex background. Therefore, according to (Luo et al., 2022), the use of a classical target detector results in reduced performance. This complexity in target detection (and, thus, target identification) may be hard to model. Unfortuna-

tely, most of the current approaches to solve a CSAT problem overlook this difficulty by introducing an idealized CVS. The capacity of the sensor to ensure the presence or absence of a target is most of the time modeled through a stochastic approach by introducing a target detection probability as well as a false alarm probability, which are often constant. Moreover, since the CVS is idealized, the target location estimation process is idealized as well and provides directly a noisy measurement of the target state as soon as it is detected within the robot sensing range. With these simplifications, the measurement noise is often modeled with a known probability density function. According to (Bai et al., 2021), this stochastic modeling of the CVS and estimators and its related assumptions are hard to impossible to justify. Moreover, such an approach is inadequate to account for the geometry of the environment when obstacles are present but unknown to the robots. Consequently, the expected performance of these CSAT strategies is only theoretical and is likely to represent an unreachable upper-bound when deployed in a real-life scenario.

Regarding the environment, it is either free of obstacles, cluttered with *a priori* known obstacles, or cluttered and unknown. On the one hand, even though the environment is known, many target search approaches only consider a 2D representation of the environment and few consider an approximated 3D representation. This choice is motivated by the fact that a 3D map of an outdoor environment is too complex to be stored in a UAV. Nonetheless, the 2D simplification comes at the cost of a new problem. If an obstacle is only approximated by a 2D element, then a UAV may be unable to evaluate the portion of its field of view that it cannot observe due to occlusions. To avoid this difficulty, one may assume that the obstacles are tall enough. Then, according to (Liu et al., 2020), the 2D approximation may be relevant if the UAV flies between the obstacles and at a lower altitude than the shortest obstacle. On the other hand, the target search in an unknown cluttered environment is still a challenging problem, especially for aerial robots since, as pointed out in (Placed et al., 2023), the task of simply mapping an outdoor 3D environment may lead to unbearable computation problems. The problem of unknown occlusions by obstacles may be solved by deploying a team of UAVs in a formation, e.g., (Ibenthal et al., 2023). Otherwise, if no mapping device is available, other approaches have to be developed to gain knowledge about the environment from the information collected by the embedded CVS in order to deploy an efficient cooperative strategy to search for targets.

To address the aforementioned problems, a set-membership approach is considered. This approach, introduced by (Schweppe, 1968), assumes that the measurement noise and other uncertainties are bounded and of known bounds. Depending of the considered problem, this assumption is as difficult to justify as the previous stochastic assumption. Nevertheless, the advantage of a set-membership approach is that nonlinear systems can be handled more easily. The Set Inverter Via Interval Analysis (SIVIA) presented in (Jaulin and Walter, 1993) is a recursive set-membership algorithm that is used to solve several inverse problems involving non-linearities. From the set-membership approach, several set-membership estimators have been proposed to solve a localization problem (Kieffer, 1999; Kenmogne Fokam, 2019; Ibenthal, 2022). More specially, the estimator developed in (Kenmogne Fokam, 2019) is, to the best of my knowledge, the first set-membership estimator that uses an image to estimate a location, here the location of the robot with the on-board camera. Additionally, the work in (Ibenthal, 2022) lays the foundation for the first method to solve a CSAT problem with a set-membership approach. However, unlike (Kenmogne Fokam, 2019), (Ibenthal, 2022) considers an idealized CVS that provides directly a noisy estimation of the location of any target that is

located within the field of view of the UAV. As mentioned, a problem of this approach is the assumed known bounds of the estimated target location uncertainty are hard to justify. The use of a set-membership approach to model the CVS, as in (Kenmogne Fokam, 2019), may allow to introduce more reliable bounds that are simpler to justify.

1.2 . Thesis outline and contributions

This thesis presents an all-in-one approach to solve the CSAT problem within an unknown structured environment by using a fleet of UAVs controlled in a distributed manner. Each UAV embeds a CVS in order to process the information collected from the portion of the environment under observation. This process of information collected by the CVS includes RGB imagery, image segmentation, depth map evaluation, and target detection/identification. Though different sensors, *e.g.*, LiDAR, stereo camera, or monocular camera, can be embedded on a UAV and different image processing algorithms, *e.g.*, R-CNN, or YOLO, can be implemented to process the information collected by those sensors, their modeling is a problem in itself. A CVS that relies on a deep-learning approach is hard to model, and the sensor model is highly dependent on the technology used. Instead, this thesis provides assumptions regarding how the CVS measurements can be interpreted. Using the pinhole model to approximate the camera, a geometric approach is used to represent the information contained in a CVS measurement. For instance, if a pixel of the acquired image is labeled Ground, then one may assume that this pixel has only been illuminated by light rays stemming from the ground, regardless of the type of camera used. The formulation of these assumptions takes advantage of the known implicit bounds of the provided measurements, *i.e.*, the fact that they are related to a pixel of known dimensions. This motivated the use of a set-membership approach to model measurement noise and other uncertainties as bounded sets with known bounds. Then, from those assumptions, a set-membership estimator is proposed to characterize, from the CVS measurements, set estimates that are guaranteed to contain the location of the targets, and set estimates that are guaranteed to contain no target location. Additionally, the CVS measurements related to the obstacles are used to create a rough approximation of the environment in order to predict what the UAV will not see in the future when it updates its trajectory. Moreover, each UAV can communicate with neighboring UAVs in order to mutually improve its knowledge about the environment or the possible locations of the targets. The structure and contributions of the thesis are detailed hereafter.

Chapter 2 reviews some work related to the problem addressed by this thesis. The CSAT problem and its challenges are presented. Some sensors and algorithms that provide CVS measurements are discussed. Several target location estimators are presented. The methods to search for a target, pursue a target, and keep track of its location when the identified target is not under observation are also discussed as well as some methods to map an unknown environment. Lastly, some control schemes are presented.

Chapter 3 formalizes the considered problem. This chapter introduces, in a first time, assumptions regarding the structure of the unknown environment. Then, the targets are modeled and their characteristics and behaviors that are assumed *a priori* available to the UAVs are detailed. The same is done with the UAVs. The camera model based on a pinhole model is presented and followed by several assumptions regarding the interpretation of the CVS measurements (pixel label, bounding box

related to an identified target, and depth map). These assumptions, even though they do not aim to model the complexity of the embedded CVS, are instrumental in defining a measurement function that explains how the acquired CVS measurements are obtained from the portion of the environment monitored by the UAV.

Chapter 4 presents - to the best of our knowledge - the first target location estimator that directly exploits CVS measurements using a set-membership approach. Target-related CVS measurements refer to pixels labeled as *Target*. Then, based on the assumptions and the geometric approach detailed in the previous chapter, the estimator characterizes a set containing all the possible points that may be related to the identified target and, thus, may have contributed to the illumination of those pixels. A strength of the set-membership approach is when the assumed bounds of the measurement noises and other uncertainties are satisfied, then one can prove that this set of points is guaranteed to contain points belonging to the target. Nonetheless, we have no interest in estimating the location of a random point belonging to an identified target. Instead, we want to estimate the location of a specific and invariant point of the target that is denoted as its location. For that purpose, we assume that the main dimensions of the identified target are *a priori* available to the UAVs. Consequently, from the set that is proved to intersect the target, another set is characterized such that it is guaranteed to contain its location. Such a result is obtained even if the identified target is partially hidden behind an obstacle. Similarly, and under some assumptions that are detailed in the previous chapter, the estimator exploits CVS measurements related to the obstacles and the ground to characterize other set estimates that are guaranteed to contain no target location. Finally, through a prediction-correction process, the set estimates obtained from the CVS measurements are used to update :

- the sets that are guaranteed to contain the location of each identified target
- the set that is guaranteed to contain the location of not yet identified targets

This update step may involve other UAVs if they are within communication range and can broadcast their set estimates. Lastly, the chapter presents our approach to update from CVS measurements an occupancy-elevation map that is used as our rough approximation of the shape of the obstacles. The simulation results presented in this chapter are only done to evaluate the performance of the proposed target location estimator. Therefore, no control design is considered.

Chapter 5 presents several cooperative control designs, all based on a model predictive controller. They aim to solve an optimization problem defined as the search for the best new list of trajectories allowing the whole fleet to a better cooperative navigation within the environment to search and track the targets. A criterion reflecting this objective has been proposed for each control design. The main difference between those criteria is the amount of knowledge that a UAV has about the environment during the trajectory update. Three different degrees of available environmental knowledge are considered :

- No environmental knowledge. The UAV is unable to gain any knowledge regarding the environment
- No memory regarding the shape of the obstacles. The UAV approximates the shape of the obstacles that are currently under observation and uses this knowledge to look for a better search and tracking trajectory, but no memory is allocated to store such a knowledge over time
- Mapping of a 3D environment. The UAV embeds the required devices, algorithms, and memory to update a map of the environment that approximates the shape of the obstacles in order to

predict during the trajectory update how they would limit its field of view.

The simulation results of this chapter compare the performance of each control design in terms of map exploration or reduction of estimated target location uncertainty, while using the target location estimator presented in the previous chapter.

Chapter 6 takes into account the possibility that the state vector of the UAVs is uncertain. This uncertainty may come from the fact that the UAV dynamic is not perfectly known, and the measurements provided by embedded proprioceptive sensors are corrupted by a bounded noise. Therefore, the uncertainty regarding the UAV location and attitude directly affects the target location estimator presented in Chapter 4. This chapter presents an adaptation of how the set estimates introduced in Chapter 4 are characterized in order to retain their useful properties even with an uncertain UAV state. Since some of the previously introduced set estimates are 3D, an interval-based approach is considered. Nonetheless, it is shown that the classic interval analysis is insufficient to efficiently address the problem of characterizing 3D set estimates from CVS measurements while accounting for UAV state uncertainty. Instead, thick-intervals are used. The simulations of this chapter reuse the simulation setup of Chapter 4 in order to illustrate the drop in performances for several examples of UAV state uncertainties.

Chapter 7 concludes the thesis, and Chapter 8 presents some research directions that come out of our work such as :

- the addition of an unknown uneven ground
- the relaxation of the assumption regarding the structured environment
- the addition of CVS errors like an imperfect target detection/identification

1.3 . Publications

International workshop :

- M. Zagar, M. Kieffer, H. Piet-Lahanier, and L. Meyer. "Robust 3D target localization using UAVs with state uncertainty". 13th Summer Workshop on Interval Methods. 2022.

Published international conference paper :

- M. Zagar, M. Kieffer, H. Piet-Lahanier, and L. Meyer. "Robust collaborative 3D target localization using UAVs with state uncertainties", in Proc. 22nd IFAC Symposium on Automatic Control in Aerospace. 2022. Vol. 55, no 22, p. 230-235.

Accepted international conference paper :

- M. Zagar, L. Meyer., M. Kieffer, and H. Piet-Lahanier. "Target Search and Tracking in an Unknown Environment by a Fleet of UAVs using a Set-Membership Approach". Accepted by Conference on Decision and Control. 2024.

Submitted international journal article :

- M. Zagar, L. Meyer., M. Kieffer, and H. Piet-Lahanier. "Set-membership target search and tracking using cooperating UAVs equipped with vision systems". Submitted to IEEE Transactions on Systems, Man and Cybernetics : Systems.

Chapter 2 - State of the art

The aim of this thesis is to design a strategy to control a fleet of Unmanned Aerial Vehicles (UAVs) such that they explore a possibly unknown environment containing obstacles to find and track an unknown number of targets. The UAVs have to find targets and estimate their locations using embedded sensors that provide measurements related to parts of the environment. The control design of the robots has to meet a compromise between tracking already identified targets and exploring parts of the environment that may contain targets not detected yet.

Section 2.1 overviews the target search and tracking problem and the different approaches to address it. Then, the sensors and computer vision algorithms that can be embedded on a robot are presented in Section 2.2. Using the measurements provided by these sensors and algorithms, the robot may update its knowledge regarding the environment by building a map using an approach presented in Section 2.3. Otherwise, the robot can exploit these measurements to estimate where the targets can or cannot be by using an approach presented in Section 2.4. Then, Section 2.5 describes some approaches to address the search and tracking problem. Finally, Section 2.6 presents some control design approaches.

2.1 . Target search and tracking using mobile robots

The efficient exploration of an environment to search, identify, and track targets is an interesting and attractive problem due to its numerous applications. This problem, for instance, encompasses monitoring of a perimeter (Liu et al., 2022b), evasive target search (Olsen et al., 2022), or exploration of hazardous areas (Zhen et al., 2020). The deployment of robots, also called *mobile sensors* or *agents* in some surveys (Robin and Lacroix, 2016; Queralta et al., 2020), to supplement or replace human operators has proved to be an efficient approach to address this kind of problem. This is due to the efficiency of the robots to evolve within a Region of Interest (RoI), e.g., underwater using submarine robots or in urban environments using aerial drones. They can reach interesting locations to collect useful information while keeping the human operator safe. Nevertheless, the deployment of robots and the design of strategies to address the target search and tracking problem depend on the mission, the available resources, the available knowledge about the RoI or the targets, and the considered models such as the target dynamic or the sensors embedded in the robots. The survey (Lyu et al., 2023) presents a variety of recent results regarding the target search and the exploration of an area after a disaster. For surveys with a broader scope, one may be interested in (Robin and Lacroix, 2016; Queralta et al., 2020).

Regarding the mission and the available resources, different classes of problems have been pointed out in (Queralta et al., 2020). Section 2.1.1 presents these classes and details their objectives and assumptions. Section 2.1.2 focuses on the problem considered in this thesis : the cooperative target search, acquisition, and tracking problem. Then, depending of the available resources (e.g., robots, sensors) and the available computing power, different levels of knowledge can be gained and different strategies can be deployed. The target search and tracking problem may be complex to address at

first glance. Thus, Section 2.1.3 decomposes the target search and tracking problem into several sub-problems that are easier to handle.

2.1.1 . Mission classification

The multi-robot target search and tracking problem encompasses a wide variety of situations and applications which, according to (Khan et al., 2016; Queralta et al., 2020), can be grouped into four main categories : Cooperative Tracking (CT), Cooperative Multirobot Observation of Multiple Moving Targets (CMOMMT), Cooperative Search, Acquisition, and Tracking (CSAT), and Multirobot Pursuit Evasion (MPE). The main differences between them rely on the knowledge available to the human operator about the targets, and on how many robots he/she has access to. For instance, the *a priori* knowledge regarding the targets may be their number, their location, or how they interact with the robots, *i.e.*, they can be neutral, cooperative, or evasive. The *a priori* available knowledge regarding the robots is their dynamics, the sensors and algorithms they embed, their communication capability, their number compared to the number of targets, *et cetera*.

Assume the number of targets and their estimated locations are known. A Cooperative Tracking (CT) mission consists in deploying the robots to pursue the known targets. This type of mission is often linked with a task of monitoring or perimeter surveillance. The number of known targets is not constrained. Therefore, if it exceeds the number of robots, then the deployed strategies have to minimize the time between two observations for each target, *e.g.*, by maximizing the visibility of the targets (Kim and Crassidis, 2010) or, minimizing the target location uncertainty (Haugen and Imsland, 2015), in order to know when a robot has to switch between targets. Otherwise, some robots may be - directly (Liu et al., 2022a) or indirectly (Zhou et al., 2021) - partitioned into sub-fleets to pursue a target from different angles. Hereafter, the notion of *target pursuit* means that a mobile robot should follow a target in order to keep it within sensing range, whereas the notion of target tracking refers to the deployment of algorithms and strategies to estimate the state of a tracked target even if it is not always observable.

When no *a priori* knowledge about the state vector of the targets is available, *i.e.*, the vector containing the target location, orientation, or other variables of interest, the mission involves a target search. The Cooperative Multirobot Observation of Multiple Moving Targets (CMOMMT) approach introduced in (Parker and Emmons, 1997) is a find-and-pursue approach where the robots switch permanently from a search mode to a pursuit mode as soon as a target is detected within their field of view. The aim is to maximize the number of monitored targets at every time instant while having a limited exchange of knowledge between the robots. This approach is well-suited when there are fewer targets than robots, see (Khan et al., 2016; Liu et al., 2022b) and the references therein. Nevertheless, the CMOMMT approach is not well-suited when the number of targets is unknown and may outnumber the robots while having a RoI cluttered with obstacles. The reason is when a robot is in the pursuit mode, it is forbidden to leave the target that it is tasked to pursue in order to explore a neighboring area that may contain undetected targets.

In Cooperative Search, Acquisition, and Track (CSAT) approaches, robots continuously solve a trade-off between searching for new targets and maintaining a good accuracy of the estimated state of those already identified. Therefore, the main difference between CMOMMT and CSAT approaches is that CMOMMT algorithms maximize the number of currently monitored targets, and thus mini-

mize the estimation uncertainty of the pursued targets, whereas CSAT algorithms try to maximize the number of identified targets while maintaining the estimated state of the already identified target as accurate as possible. For instance, (Dames, 2020) proposes a CSAT algorithm where targets can freely leave and enter the RoI. When the RoI is an area containing targets to find after a disaster, the term Search And Rescue (SAR) is preferred, see (Lyu et al., 2023) and the references therein.

The above approaches assume that the targets either cooperate with the robots to help find them, or are neutral, or move slower than the robots. When the targets are faster or are evasive, *i.e.*, the targets try to avoid being observed, Multirobot Pursuit Evasion (MPE) approaches have been developed. In (Vidal et al., 2002; Olsen et al., 2022), MPE algorithms use game theory (Basar and Olsder, 1998) or rely on other strategies such as auction mechanisms (Cai et al., 2008) to design trajectories minimizing the time needed to either physically or visually capture a target.

Other categories of problems exist such as static surveillance, also named the *art gallery problem* (O'Rourke and others, 1987), when the robots are static. When the targets are static, then tracking/pursuit steps are not required anymore. Instead, the target search in this scenario is rather close to an area-covering problem as in (Kuhlman et al., 2017). In addition, one may remark that, as soon as all targets located within the RoI are identified and their state is estimated, the initial CMOMMT or CSAT/SAR problem becomes a CT problem.

2.1.2 . Cooperative Search, Acquisition, and Track

Unlike CT approaches, which assume to already know where the targets are, or MPE approaches, which consider specific target behaviors, CSAT/SAR approaches can be related to a wide variety of real-life problems and applications. See (DJI,) only as an example of a list of situations that can be encountered in the context of a SAR problem. Traditionally, the CSAT approach only considers mobile targets whereas the SAR approach may consider either static or mobile targets. For the latter, the main difficulty of the SAR approach is to deal with an unknown and unstructured RoI, which is the case after a natural disaster (storm, earthquake). In such a situation, the targets do not or are unable to move. In the case of a marine rescue, the target may drift away. For the rest of this section, we will only consider mobile targets for a target search and tracking problem with CSAT/SAR approach. See (Khan et al., 2016; Lyu et al., 2023) and the references therein for CSAT and SAR algorithms respectively.

As for CMOMMT algorithms, CSAT algorithms consider two modes : a search mode where the robot explores the RoI to find a target, and a pursuit mode where the robot pursues a target to collect more information in order to reduce its state estimation uncertainty. Unlike CMOMMT, the robots controlled by a CSAT algorithm will not be locked in pursuit mode once they have found a target. Instead, CSAT algorithms are designed to enable a robot to switch between the two operating modes. CSAT algorithms have been first introduced in (Furukawa et al., 2006) : Four UAVs are deployed to search for six mobile ground targets. The UAVs are able to communicate with all the other UAVs without failures and delay. They exchange information to mutually update their knowledge regarding the actual possible locations of the targets. Then, each robot individually evaluates its best sequence of control actions which maximizes an objective function involving both the search and the pursuit part. The search mode is designed to explore the RoI in order to reduce the uncertainty of the target location, modeled with a probability density function. A robot switches from search mode to pursuit mode if it has detected a target. Then, it switches back to search mode when the target is considered

rescued, meaning that a rescuing helicopter had the time to reach the actual location of the pursued target to retrieve it. Nonetheless, most of the other CSAT problems do not involve a vehicle that will remove the pursued target from the RoI. Therefore, as in (Frew and Elston, 2008; How et al., 2009), the robots explore the RoI to acquire new observations in order to estimate where the targets can or cannot be. A robot switches from search mode to pursuit mode to reduce the estimated state uncertainty of a detected target. Then, it switches back to search mode if the uncertainty of the pursued target estimated state is below some threshold. Thus, the robot stops monitoring the targets and starts searching for new ones. Nevertheless, since the targets move and their dynamics are at least partially known, their estimated state uncertainty grows until they are re-observed by a robot in the pursuit mode.

This threshold, and more generally the transition condition between search mode and pursuit mode, controls the trade-off between searching for new targets and recapturing already identified targets to reduce the uncertainty regarding their estimated state. An example of transition condition is described in (Frew and Elston, 2008). The spectral norm of a target state covariance matrix is used as the metric to evaluate the uncertainty of its estimated location. The evolution over time of this uncertainty can be predicted. Therefore, as long as a robot has the time to recapture the target before an uncertainty limit is violated, it remains in search mode. In (How et al., 2009), the mode transition is managed with a task assignment between the UAVs. However, as highlighted in (Ibenthal, 2022) the design of the transition condition has to be defined carefully to avoid the estimated state uncertainty of the identified and non-detected targets growing too fast. Recently, a new approach has been proposed to simultaneously search and pursue targets (Ibenthal et al., 2020; Banerjee and Schneider, 2024; Li et al., 2024). While exchanging information with its neighbors, a robot searches for the best action it can take to minimize a criterion that accounts simultaneously for the future gain of exploring areas potentially containing undetected targets and the future gain in recapturing already identified targets to get a more accurate estimation of their states.

2.1.3 . Problems related to the target search and tracking problem

A target search and tracking problem involves several disciplines - or subproblems - such as the processing of the information collected by the embedded sensors, a robust target state estimation, or the design of a controller. For instance, a robot embeds sensors and computer vision algorithms to gain new measurements regarding the targets or the environment. Measurements related to the presence of targets are used to estimate their states (Meera et al., 2019), whereas measurements related to the absence of targets are used to remove some of their potential locations and, thus, reduce the uncertainty of the estimated target location (Blanding et al., 2006). Measurements related to the environment are used to gain some knowledge such as the location of obstacles or the characterization of pathways (Zhang et al., 2022). This kind of knowledge is dubbed as environmental knowledge. Such knowledge may take the form of a 2D or a 3D map used as input for exploration or target search/pursuit strategies. Exploration strategies aim to drive the robot toward unexplored areas of the RoI for mapping purposes (Hardouin et al., 2020). Target search strategies aim to drive the robots toward areas that may contain either a new target to detect or a target to recapture (Tang et al., 2019). Pursuit strategies aim to evaluate new observation locations to further increase the accuracy of the estimated state of currently monitored targets (Cichella et al., 2015; Ibenthal et al., 2021).

To summarize the inherent complexity of a target search problem, this thesis proposes the following subproblems :

- *Acquisition of information* : the embedded sensors collect new information from the monitored environment. Then, the collected information is processed using computer vision algorithms
- *Extraction of knowledge* : interpretation of the measurements provided by the sensors and the computer vision algorithms to gain knowledge about the possible values of the target state as well as the environment through an estimation process
- *Target search* : application of strategies to continuously collect new information about where the targets can and cannot be
- *Target pursuit* : application of strategies to keep a target within sensing range
- *Target tracking* : exploitation of *a priori* knowledge about the targets to predict their evolution from previously gained knowledge in order to, *e.g.*, evaluate the next best location for re-observation
- *Map exploration* : use of either *a priori* or gained environmental knowledge to design an efficient robot trajectory within the cluttered area in order to optimize the future gain of information
- *Cooperation* : deployment of a network architecture (centralized, decentralized, or distributed) to allow the robots to cooperate, *i.e.*, exchange knowledge, and thus, increase the overall performance of the proposed approach.

These subproblems and the current approaches to solve them are discussed in what follows.

2.2 . Sensors and computer vision algorithms

The first difficulty regarding the target search and tracking problem is how to collect new information from the perceived environment and exploit this information to infer the possible values of the state of each target contained in the RoI. To achieve this goal, the robots embed Computer Vision Systems (CVS) composed of sensors (there is in general at least one camera) and computer vision algorithms to process the information collected by those sensors. Alternatively, the robots may transmit the acquired sensor measurements to a robot or a ground unit with a higher computing power, or to store them for offline processing.

In what follows, we distinguish the information contained in a scene that is under observation, the measurements provided by the CVS, and the knowledge acquired by the robot. What we call information is - to put it simply - all that composes an environment. For instance, a building is not only defined by its shape but by its color and location too, or a target can be defined by a heat signal or its features such as the license plate for a vehicle. The embedded sensors collect some of this information to provide raw measurements such as an image, but this collected information can be processed by embedded computer vision algorithms to provide other measurements such as a depth map or the detection of a target. In what follows, a measurement provided by either the sensors or the computer vision algorithms is dubbed as CVS measurement. Finally, the available CVS measurements are processed further by an estimator to gain knowledge on what the sensors have perceived : target location, absence of target, presence of an obstacle, *etc.* Figure 2.1 illustrates this chain to get new knowledge from the portion of the environment that is under observation.

Section 2.2.1 presents some examples of sensors. Section 2.2.2 presents some computer vision

algorithms that can be used with some of the aforementioned sensors. Lastly, Section 2.2.3 discusses the accuracy of some computer vision algorithms.

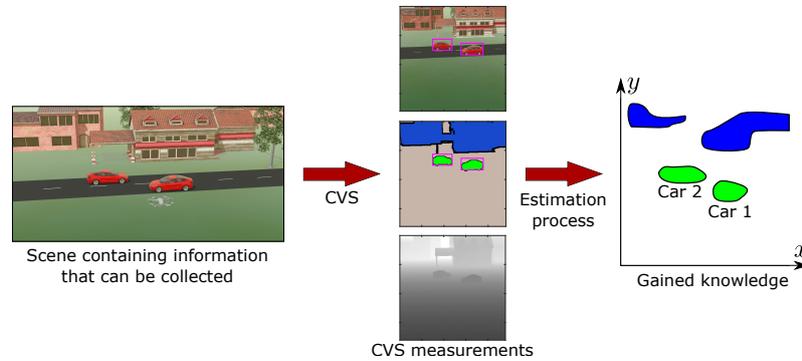


Figure 2.1 – Illustration of the chain to get new knowledge from an observed scene. A UAV is monitoring an area with cars and obstacles. Its embedded CVS provides new measurements such as an image, bounding boxes (purple) of the moving objects, a segmentation, or a depth map. Then, these CVS measurements are used to estimate the location of the cars (green sets) and the location of the obstacles (blue sets).

2.2.1 . Examples of sensors

Exteroceptive and proprioceptive sensors are two categories of sensors that can be embedded in a robot. Proprioceptive sensors provide measurements of the state of the robot on which they are embedded whereas exteroceptive sensors provide measurements related to the environment. This section only presents exteroceptive sensors.

In most cases, a robot embeds an RGB camera to get images of parts of its environment. The camera may be supplemented by a LiDAR or a sonar to get additional distance measurements to surrounding obstacles to get a depth map. As in (Williams et al., 2020), a LiDAR can be used to get an approximation of the environment through a point cloud, *i.e.*, a discrete set of points in space used as a way to approximate the surface of objects and structures. Figure 2.2 provides a result regarding this mapping technique. Alternatively, a depth map can be obtained using a stereo camera (Park and Kim, 2012) through the computation of a disparity map or a monocular camera using structure-from-motion algorithms (Ranftl et al., 2016). The latter require, however, multiple images from the same scene and are more time-consuming (Madhuanand et al., 2021). More recently, deep-learning techniques have been proposed to compute depth estimates from monocular images. (Madhuanand et al., 2021) proposes a self-supervised depth estimator trained using a dataset containing videos (RGB and depth map) generated by a UAV flying in an urban environment at an altitude ranging from 50 to 100 meters. (Shimada et al., 2023) proposes a similar approach with a combination of a CNN and an Optical Flow algorithm, trained with a dataset generated by AirSim. Lenslet light field cameras can also be used to generate a depth map as in (Jeon et al., 2015) where the authors propose a stereo-matching algorithm that has been modified to account for the angular coordinates of the light field.

Other sensors may be considered depending on the context, like in (Ji et al., 2022) with a gas sensor, or in (Van Tilburg, 2017) with a thermal camera, but their utilization will not be considered in this thesis.

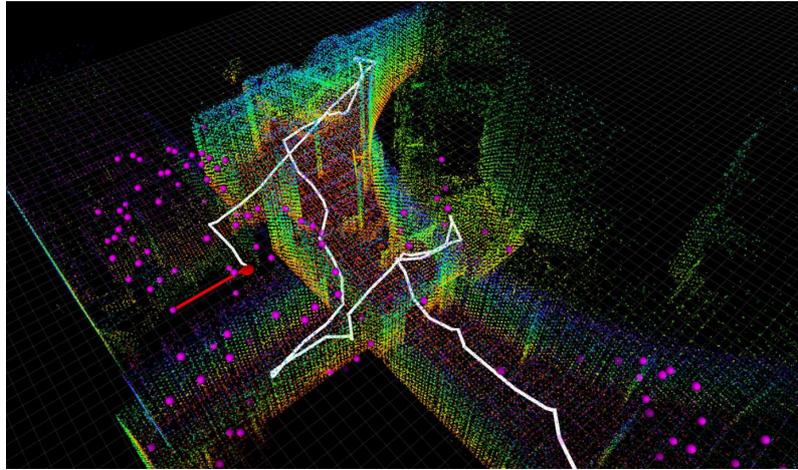


Figure 2.2 – Snapshots of the exploration mission presented in (Williams et al., 2020). The magenta dots are candidate locations that the robot may choose to reach to continue the exploration. The white line is the trajectory that the robot took. The red line shows the path to the selected candidate location. The point cloud represents the portion of the indoor environment that has been mapped.

2.2.2 . Examples of computer vision algorithms

Using computer vision algorithms, other information - dubbed as measurements in what follows - may be obtained from the information collected by the sensors. Image or point cloud segmentation (Grilli et al., 2017; Howard et al., 2019; Minaee et al., 2021) can be used to evaluate the category of the objects perceived in the environment. Then the approach presented in (Jiang et al., 2023) can be applied to improve image segmentation using the confidence that each pixel has in belonging to a class. Regarding target detection, one may rely on deep-learning methods such as R-CNN (Girshick et al., 2014), SSD (Liu et al., 2016), or YOLO (Redmon et al., 2016; Luo et al., 2022; Jiang et al., 2022). Figure 2.3 provides some results for a target detector embedded on an aerial robot that has been presented in (Luo et al., 2022) where several vehicles are localized in the image using boxes. In (Arras et al., 2007), the authors use a point cloud to detect the presence of a human target by detecting its legs. (Minaeian et al., 2018) propose another target detector specially designed for the low computing power of UAVs, which combines optical flow for feature tracking with background elimination.

Moreover, depending on the application and the available dataset, more complex measurements can be obtained such as the current action of a human (Shinde et al., 2018), *e.g.*, handshaking or phone call. Another example of computer vision algorithms is the 6D-pose detection that relies on a deep-learning approach. These algorithms can provide an estimation of the position and the orientation of a static object (Atanasov et al., 2014; Zeng et al., 2018), or a vehicle (Sahin et al., 2020; Fan et al., 2022). Nonetheless, these algorithms often require strong *a priori* information such as 3D models of the target shapes.

Target identification, *i.e.*, the capacity to estimate the identity of a target, strongly depends on the nature of the target. Some approaches, such as (Goldhoorn et al., 2018; Zhu et al., 2021; Liu et al., 2022c), use QR codes or ArUco markers attached to the target to simplify their identification. Without such markers, one can use the Bi-directional Similarity Matching algorithm, presented in (Oron et al., 2017)



Figure 2.3 – Illustration of the detection results obtained with the YOLOD algorithm developed in (Luo et al., 2022). Red boxes are for detected cars. Purples boxes are for detected pickups. Orange boxes are for detected trucks. Cyan boxes are for detected boats. Green boxes are for detected camping cars.

and used in (Liu et al., 2022a), to detect a target, store in memory information regarding its shape, and try to re-identify the target in the future by testing if the newly collected information matches with that stored, see Figure 2.4. The authors of (Silva et al., 2019) consider a similar approach, but it only works with the extracted features of a human face.

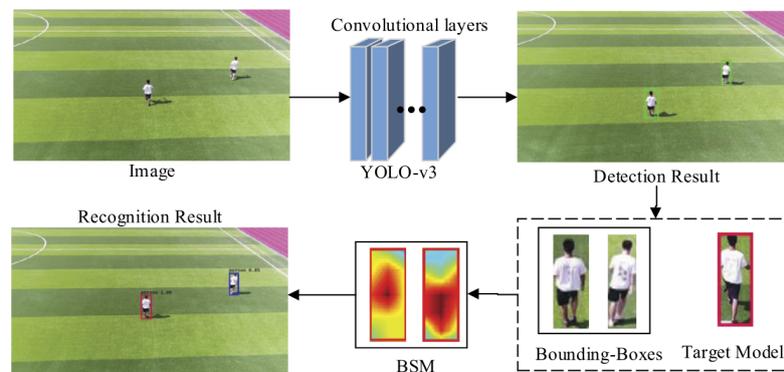


Figure 2.4 – Illustration from (Liu et al., 2022a) of the detection of human targets using YOLOv3 and the identification of the correct target to track with the Bi-directional Similarity Matching (BSM) algorithm (Oron et al., 2017).

2.2.3 . Accuracy of computer vision algorithms

Most of the aforementioned CVS are, nowadays, commonly embedded in robots to collect and process new information before applying estimation and control design algorithms with the newly obtained CVS measurement. Nevertheless, the measurements provided by these CVS are not perfect, see (Jamwal et al., 2016) as a survey of the accuracy of UAV-based mapping approaches using structure-from-motion. Outliers, e.g., the detection of a target that is not present or the non-detection of a

visible target, can occur and have to be correctly managed to avoid failures in the following processes. Though the generation of outliers is rare for some computer vision algorithms, it is still useful to have access to some metrics evaluating their accuracy to know if robust approaches have to be considered in order to reject those outliers.

Some metrics have already been introduced for target detection. To evaluate the accuracy of the target detection in one picture, one may use the Intersection over Union (IoU). The IoU represents how accurate is the localization of a detected target in the image. The way this metric is evaluated is by comparing the box representing the ground truth, *i.e.*, what the target detector is aimed to achieve, with the box representing where in the image a target has been detected. Figure 2.5 provides an example. Then, to get an estimation of the expected accuracy of a target detection algorithm, the mean average precision (mAP) may be used. The mAP is obtained by computing the average over the number of considered categories (car, bus...) of the average of the precision for each considered category. The precision is defined as the ratio between the true positives, *i.e.*, having an IoU above some threshold, and the sum of the number of true positives and false positives, *i.e.*, detection of a target that does not exist. Another metric is the F1 score. It is the harmonic mean of the precision and the recall of the sensor. The recall is the ratio between the true positives and the sum of the number of true positives and false negatives, *i.e.*, non-detection.

In (Luo et al., 2022), the mAP of their algorithm designed to detect targets from aerial images is 83.19 %, which is better than YOLOv4 (Bochkovski, 2020) (with an mAP of 80.13 %) or Faster R-CNN (Ren et al., 2016) (with an mAP of 44.39 %). Nonetheless, the probability of bad target detection still exists and the bounding box provided by the embedded target detector may not or only partially contain the target.

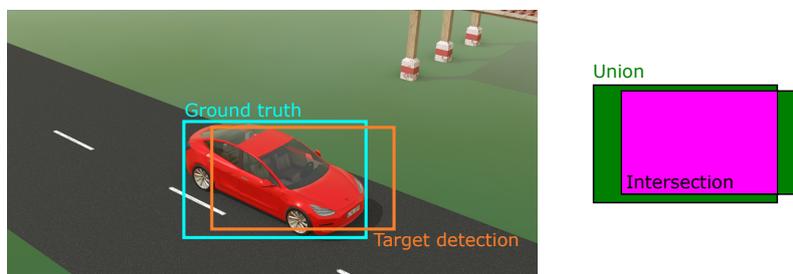


Figure 2.5 – Illustration of the IoU metric. A car is visible the image. The blue box is the ground truth, and the orange box is what the target detector (trained to detect cars) returns. The IoU is the ratio between the intersection between the two boxes and the area of their union.

Other highly inaccurate measurements may be provided by the CVS when, *e.g.*, target misidentifications or non-detection occur. These outliers are commonly referred to either false positives or false negatives and can make the entire estimation process fail. (Symington et al., 2010) and (Meera et al., 2019) have proved that the quality of a deep learning target detection algorithm depends, among others, on the observation condition (including the presence of obstacles) and weather conditions. This is due to the fact that the performance of these algorithms heavily depends on the content of their training dataset. If the dataset is not consistent enough to encompass all possible situations encountered by the robot, the provided measurement is biased because it is only obtained after a

comparison between what has been actually perceived and with what the deep-learning algorithm has been trained. Consider the 6D-pose detection algorithm proposed in (Mousavian et al., 2017) as an example. Though the average precision of their approach is 92.98 % , their algorithm can still fail to provide a correct estimation of the target location and dimensions, as illustrated in Figure 2.6.

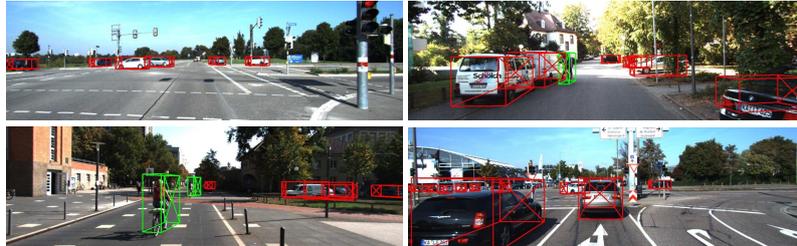


Figure 2.6 – Illustration of the performance of a 6D-pose algorithm. Red boxes are for cars and green boxes are for cyclists. Most of the estimated locations, orientations, and dimensions are correct. Nonetheless, in the top-right figure, the algorithm provided an erroneous pose estimation of the partially-visible black car. Figure taken from (Mousavian et al., 2017).

The measurements provided by the CVS embedded on a robot can be either used to map the environment or to characterize areas where a target can or cannot be.

2.3 . Map building

A target search problem involves the navigation of a robot within an environment cluttered with obstacles whose locations are often unknown. This section considers the problem of gaining knowledge about the location and shape of obstacles in some unknown environment using CVS measurements. To achieve this goal, several approaches to map the environment have been proposed in the context of Simultaneous Localization and Mapping (SLAM) algorithms, see the survey (Placed et al., 2023). Their differences are mainly related to the choice of the representation used to describe the environment, *i.e.*, with a point cloud or with a grid map.

A first representation of the environment is by point clouds, *i.e.*, a set of 3D points approximating the surface of objects. This type of representation is well suited when distance measurements are acquired by a LiDAR sensor. An example of a point cloud used for the mapping of a city is provided in Figure 2.10. The point cloud is build progressively. The main difficulty is to put in correspondence newly acquired measurements with previous knowledge about the point cloud or the model representing the environment. Iterative Closest Point (ICP) algorithms (Besl and McKay, 1992) compare two point clouds : one is the *data* and the other is the *model*. The aim is to find the best transformation that makes the *data* point cloud match the *model* point cloud. From this method, approaches to iteratively build a map of the environment have been developed. (Tomono, 2009) uses a stereo camera to generate a point cloud to map a 3D indoor environment. An approach adapted to outdoor environments is presented in (Mendes et al., 2016). The authors propose a graph approach in which each node is related to a scan, *i.e.*, a point cloud. For each newly obtained point cloud, if it does not overlap too much with previous scans, a new node is created to store the scan. To evaluate the overlap, the scan is not compared with the entire registered map but only with a local map built from previous

scans. Nonetheless, to reduce the computational burden of ICP, up to 80% of the dense point cloud is filtered without losing the structural information of the obstacles. Alternatively, (Williams et al., 2020) use a grid approach to limit the number of registered 3D points

A second representation of an environment uses a 3D occupancy grid. The authors of (Hardouin et al., 2020) propose a method to cooperatively reconstruct the surface of a building using an occupancy grid. An occupancy grid consists of cells which are either *unknown* and need to be explored, *empty* and contain no obstacle that may obstruct the path of the robot, or *occupied* meaning that at least a part of an obstacle or an object is located in the cell. (Serdel et al., 2023) present two different approaches to build these 3D grid maps, *i.e.*, by using either Octomap (Hornung et al., 2013) or Voxelblox (Oleynikova et al., 2017). (Asgharivaskasi and Atanasov, 2023) use a semantic Octomap representation to explore and map an unknown outdoor environment, see Figure 2.7. The semantic segmentation has been performed using a neural network that takes RGB images as inputs. See (Andersone, 2019) and the reference therein for other examples of grid maps.

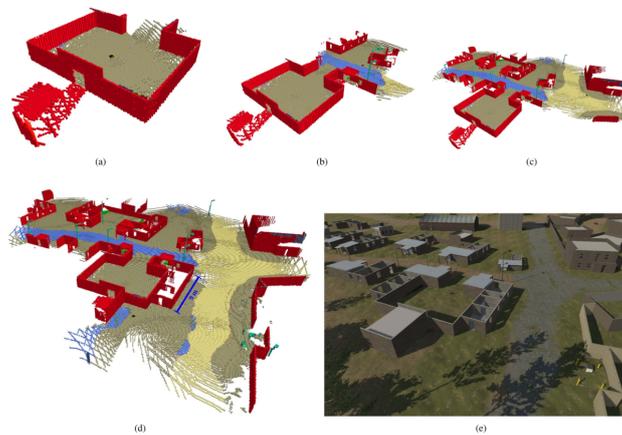


Figure 2.7 – Figure from (Asgharivaskasi and Atanasov, 2023) illustrating the grid-based representation (via semantic Octomap) of the RoI during its exploration.

The availability of semantic information, such as classification (Grilli et al., 2017), *i.e.*, knowing what class of object has been detected, or traversability (Stelzer et al., 2012; Liu et al., 2023), *i.e.*, evaluating the capacity of a robot to cross a mapped terrain, can improve the mapping algorithms (Atanasov et al., 2014; Zhong et al., 2018; Zhang et al., 2022; Asgharivaskasi and Atanasov, 2023; Serdel et al., 2023) by, *e.g.*, distinguishing static and mobile objects as in (Zhong et al., 2018). To map an outdoor environment, (Zhang et al., 2022; Serdel et al., 2023) use semantic information to evaluate the traversability between two locations.

Nevertheless, as pointed out in (Placed et al., 2023), 3D mapping and exploration of large outdoor environments is still challenging, especially for aerial vehicles due to their limited computational power when they are tasked to map a city-scale environment with a 3D occupancy grid. Using an Octomap may reduce the complexity as shown in (Asgharivaskasi and Atanasov, 2023), but their study is limited to the ground exploration of an area of 22500 m² (equivalent to a 150 m × 150 m²).

For the mapping of a large-scale outdoor environment, alternative 2.5D mapping approaches represent less accurately the environment in the third coordinate, *i.e.*, the height. This approximation

allows the use of simpler geometric representations of the environment at the cost of a decrease in the quality of the description of complex structures and objects of the RoI. An example is the use of an elevation map. This approach consists of a 2D grid where each cell contains the value of the measured floor height. In (Fankhauser et al., 2014), the elevation map is updated from the noisy measurement provided by a ground robot, whereas a UAV is used instead in (Zhang et al., 2022). The authors of (Du et al., 2018) present a method to get a bounded elevation map with a set-membership approach. Moreover, (Zhang et al., 2022) use, among others, an elevation map with a traversability map, see Figure 2.8, to rapidly map the RoI with a UAV in order to provide enough information for a ground vehicle to find a trajectory that crosses the previously unknown RoI to reach a known target location.

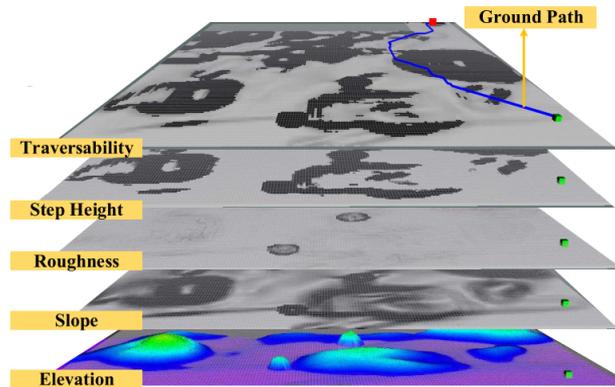


Figure 2.8 – Figure from (Zhang et al., 2022) illustrating the maps that are considered during the exploration and mapping of the environment using a UAV. With the elevation map, other properties of the terrain are evaluated : the slope, the roughness, and the step height. These metrics are evaluated using (Stelzer et al., 2012). Once obtained, the traversability map is deduced.

(Dellenbach et al., 2022) also uses an elevation map to solve a SLAM problem in an outdoor environment where, instead of applying the ICP algorithm to the newly obtained point cloud, they transform the point cloud into an elevation map and then search the 2D rigid transformation between the newly obtained elevation map and the stored elevation maps. When a match is obtained, an ICP refinement is performed with the initial elevation map’s point clouds for the loop closure.

Alternatively, the authors of (Zhang et al., 2023) assume that the buildings located inside the RoI can be approximated by a pyramidal stacking of parallelepipedic blocks. Therefore, the map representation is based on layers, see Figure 2.9 and Figure 2.10. This approximation enables the generation of fictive point clouds used for vehicle localization which are quite close to the ground truth.

(Souza and Goncalves, 2016; Du et al., 2018; Souza and Goncalves, 2018; Bayer and Faigl, 2019) combine the elevation map with a 2D occupancy map in order to map the RoI. An *occupied* cell is a cell that is at least partially occupied by an obstacle for which its height is either known or has been estimated using CVS measurements. An *empty* cell of the Occupancy-Elevation Map (OEM) represents an area where there is no building, and where the robots and the targets are free to move. In (Souza and Goncalves, 2016; Souza and Goncalves, 2018), at time t_k , the cell n of the OEM carries an occupancy probability, an estimated height $\mu_{n,k}$, and the variance $\sigma_{n,k}^2$ of the estimated height. The update of the occupancy probability relies on a stochastic sensor model that provides an occupancy probability

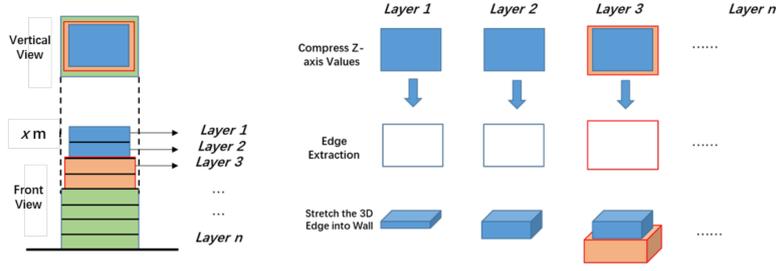


Figure 2.9 – Figure from (Zhang et al., 2023). Illustration of the assumption related to the pyramidal stacking of parallelepipedic blocks to model urban buildings.

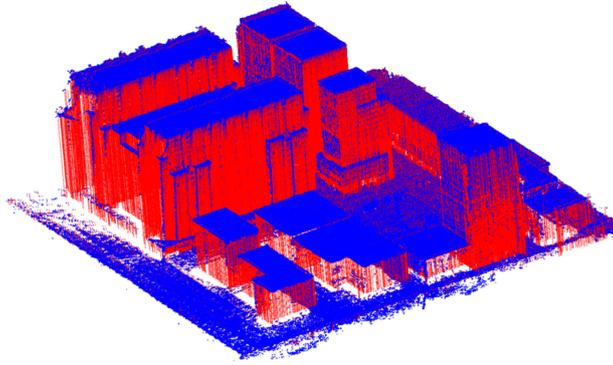


Figure 2.10 – Figure from (Zhang et al., 2023). Illustration of the building reconstruction approach with a point cloud initially obtained from CVS measurement (in blue) and the fictive point cloud generated from the pyramidal assumption (in red).

for each cell that is flown over by a ray, see Figure 2.11. The update of the estimated height and the uncertainty of the cell n is

$$\mu_{n,k} = \frac{\sigma_{n,k-1}^2 h_t + \sigma_k^2 \mu_{n,k-1}}{\sigma_{n,k-1}^2 + \sigma_k^2}, \quad (2.1)$$

$$\sigma_{n,k}^2 = \frac{\sigma_{n,k-1}^2 \sigma_k^2}{\sigma_{n,k-1}^2 + \sigma_k^2}, \quad (2.2)$$

where h_t and σ_k^2 are the newly estimated height with its uncertainty. A set-membership approach approach is used in (Du et al., 2018) to build an elevation map. Their set-membership estimator provides set estimates that are characterized by ellipsoids. From these set estimates, the authors update the elevation map. Yet, instead of simply updating the height of each cell, they propagate the bounded uncertainty of the measured height by obtaining two bounds for each cell of the elevation map, *i.e.*, the highest and the lowest elevation.

This approach to approximate the shape of the buildings in an urban environment, as in as in (Hu et al., 2019; Liu et al., 2020; Wu et al., 2020), can be supported with the main assumption of (Zhang et al., 2023) which approximates buildings as stacking of parallelepipedic blocks.

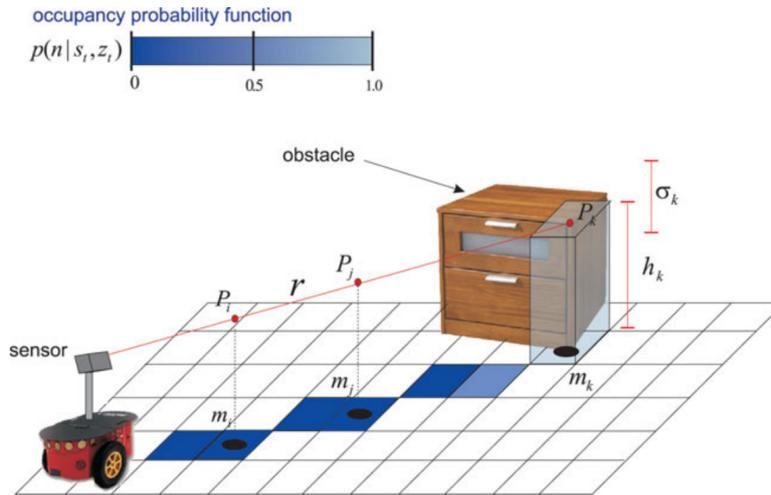


Figure 2.11 – Figure from (Souza and Goncalves, 2016). Illustration of the sensor model. A beam is cast in a direction. The cells of the OEM that are flown over are detected, and a probability of being occupied by an obstacle is assigned to them. This occupancy probability accounts for a measured distance.

Regardless of the approach used to build a representation of the environment, such environmental knowledge can be used to better estimate the location of the targets.

2.4 . Target localization

This section describes related work considering the problem of estimating the target location from CVS measurements. The difficulty and type of approaches to address this problem depend on the type of measurements provided by the CVS. It also depends on the assumptions regarding the measurement noises and other uncertainties and, thus, the approach to consider so as to account for those noises in order to process those measurements.

A deterministic approach is presented in Section 2.4.1. It is then followed by the presentation of a stochastic approach in Section 2.4.2, and a set-membership approach in Section 2.4.3. Section 2.4.4 introduces the negative information and how it can be used during the estimation process. Lastly, Section 2.4.5 discusses a simplification of the CVS that can be found in several approaches initially developed to solve a target search and tracking problem.

2.4.1 . Deterministic approach

In (Liu et al., 2022a), a UAV and a ground robot cooperate to track a human inside a RoI free of obstacles but containing other humans. The UAV and the ground robot embed a camera, the YOLOv3 algorithm for target detection, and the Bi-directional Similarity Matching algorithm for target identification. The available measurement is the bounding box provided by YOLOv3 which allows the re-identification of the target using the Bi-directional Similarity Matching algorithm, see Figure 2.4. From this bounding box, the proposed estimator implemented on the UAV exploits only its bottom-center pixel because this particular pixel is assumed representative of the target location. Then, the

algorithm estimates the target location by searching for the intersection between the flat horizontal ground and the line passing through this pixel and the camera optical center. This is achieved by using Thales theorem. Similar processes to determine the location of a target from its estimated location in the image can be found in (Minaeian et al., 2015; Vanegas et al., 2016; Sun et al., 2016; Wang et al., 2016; Liu et al., 2018). Some of these approaches also assume to know *a priori* the height of the target. Nevertheless, compared to approaches like 6D-pose estimation using a CAD model of the target, the absence of knowledge regarding the target shape may lead to relatively large and unpredictable estimation errors, as evidenced by (Di Gennaro and Waldmann, 2023) where estimating the location of humans evolving in the overlapping FoV of static cameras is achieved by stereoscopy techniques, see Figure 2.12. In addition, since these estimators are often developed for target pursuit, *i.e.*, the robot is controlled to keep the target within sight, their estimation algorithm does not provide an evaluation of the uncertainty regarding the estimated target location.

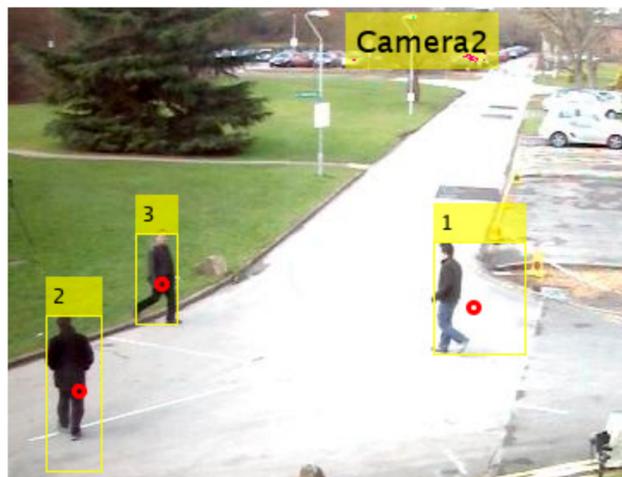


Figure 2.12 – Figure from (Di Gennaro and Waldmann, 2023) where three detected targets evolving within the overlapping FoV of static cameras have to be localized using stereoscopy techniques. Target 1 has been detected. However, their algorithm has poorly estimated its location.

2.4.2 . Stochastic approach

To evaluate the uncertainty regarding the estimated target location, one has to model the measurement noises and other uncertainties. An approach to model these uncertainties is to rely on a probability distribution.

CVS-based approaches estimating the location of a target and its related uncertainty have been proposed in (Goldhoorn et al., 2018; Meera et al., 2019). The authors of (Meera et al., 2019) consider a UAV that is tasked to find ground targets scattered within a region cluttered with obstacles. These obstacles are known and modeled using a 3D grid map. A probabilistic model combining the embedded camera and a target detector (YOLO Tiny 2.0) is used. The authors choose a normal distribution to model the sensor performance, and its parameters are chosen according to an empirical study of the accuracy of the implemented target detector illustrated in Figure 2.13. Targets are assumed to be present on a flat ground. The field of target occupancy is modeled using a discretized Gaussian pro-

cess. After an observation, the FoV of the UAV is projected on the ground to identify which cells have to be updated using a recursive technique presented in (Popović et al., 2017). This projection accounts for the part of the RoI masked by obstacles. In (Goldhoorn et al., 2018), the authors consider a ground robot, evolving in a mapped environment, which embeds a LiDAR for human detection and a camera for identification. They model what the sensors perceive with the following probabilistic model that evaluates the probability of seeing a point x_2 from a point x_1 :

$$P_{\text{visibility}}(x_1, x_2) = \begin{cases} 0 & \text{if RAY}(x_1, x_2) \text{ intersects an obstacle} \\ p_{\text{obs}} & \text{else if } \|x_1 - x_2\| < d_{\text{max}} \\ \max(0, p_{\text{obs}} - \alpha(\|x_1 - x_2\| - d_{\text{max}})) & \text{otherwise.} \end{cases} \quad (2.3)$$

$\text{RAY}(x_1, x_2)$ is the raytrace function that uses the known map to evaluate when the segment $[x_1, x_2]$ crosses an obstacle. The parameter p_{obs} represents the probability that the robot sees x_2 from its current location x_1 . The third line of (2.3) models the possible occlusions generated by nearby unknown dynamic obstacles, *i.e.*, other humans.

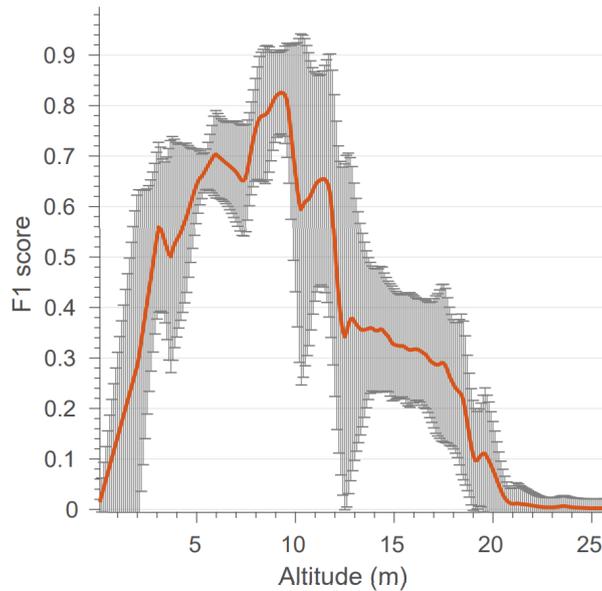


Figure 2.13 – Empirical study of YOLO Tiny 2.0 to detect the presence of a human-like target. The chosen metric to evaluate the accuracy of the target detector is the F1 score. Figure taken from (Meera et al., 2019).

2.4.3 . Set-membership approach

Instead of considering known probability distributions, the measurement noises and other uncertainties may be assumed as bounded. Several representations of these bounds exist. The author in (Schweppe, 1968), who is the first to solve a state estimation problem with a set-membership approach, used ellipsoids. An interval representation has also been introduced in (Moore, 1979; Jaulin and Walter, 1993). Unlike stochastic techniques, which provide point estimates with a confidence set,

set-membership techniques characterize a bounded set denoted as a set estimate. This set estimate gathers all the values that are consistent with a measurement model and the assumed bounds of the noises and other uncertainties that perturb the measurement process. Then, if the assumptions regarding the bounds related to these perturbations are satisfied, the set estimate provided by a set-membership estimator always contains the true value of the estimated state (Jaulin et al., 2004).

Regarding a target localization problem at time t_k , e.g., (Kieffer, 1999; Jaulin, 2009), it is assumed that the target location at time t_{k-1} belongs to a set estimate \mathbb{X}_{k-1}^t . Then, with a prediction step that will be detailed later, a set estimate $\mathbb{X}_{k|k-1}^t$ is obtained and contains all the possible locations of the target at t_k known to the robot. $\mathbb{X}_{k|k-1}^t$ can thus be considered as an *a priori*. Then, at time instant t_k , the CVS embedded in the robot provides a measurement \mathbf{y}_k of the target location. This measurement has been corrupted by a perturbation denoted by $\mathbf{w}_k \in \mathbb{W}$, with \mathbb{W} being the assumed bounded set where the perturbation can take its values. The goal is to deduce from \mathbf{y}_k , \mathbb{W} , and $\mathbb{X}_{k|k-1}^t$ the *a posteriori* set estimate containing the target location. For that purpose, a measurement function \mathbf{f} modeling the relation between \mathbf{y}_k and \mathbf{x}_k is required in order to characterize the set estimate

$$\mathbb{X}_k^t = \left\{ \mathbf{x} \in \mathbb{X}_{k|k-1}^t \mid \exists \mathbf{w} \in \mathbb{W}, \mathbf{y}_k = \mathbf{f}(\mathbf{x}, \mathbf{w}) \right\}. \quad (2.4)$$

This set estimate gathers all the possible locations of the target that are consistent with the *a priori* $\mathbb{X}_{k|k-1}^t$, the available measurement \mathbf{y} , and the assumed bounded measurement noise \mathbb{W} . Assuming the measurement function \mathbf{f} is a reliable representation of the processing performed by the CVS, and the measurement bounds are not violated, then the solution set \mathbb{X}_k^t is guaranteed to contain the actual target location at time instant t_k .

An application of this approach for the self-localization of an underwater drone in a pool is illustrated in (Jaulin, 2009). The drone is equipped with a sonar providing a measure of distance to the walls of the pool, but the measurement is corrupted by a bounded additive measurement noise. The authors in (Kenmogne et al., 2019) solve a similar problem involving a cooperative fleet of UAVs. Each UAV embeds a camera that is described by the simple pinhole model (Faugeras, 1993). The camera is used to perceive landmarks of known locations in order to estimate its location. The measurement is an image in which the known landmarks can be identified. The drones are also equipped with a ranging system antenna in order to get the distance between the drone and its neighbors, and thus get more constraints regarding the possible locations of the drone. The pinhole model is used to define the measurement function relating the available measurement to the environment. The noise related to the measurements provided by the camera depends on the resolution of the image, which is clearly bounded, but the distance measurement is also corrupted with an assumed bounded noise.

2.4.4 . Negative information

A shared characteristic of a target location estimator is that it relies on the exploitation of CVS measurements which are directly associated with the presence of a detected/identified target. These CVS measurements can be considered as True Positives when they are related to a correctly detected/identified target. In addition, one may also exploit True Negative measurements, *i.e.*, CVS measurements concluding in the (true) absence of a target in the FoV of the sensor. The exploitation of these True Negative measurements, or negative information as introduced in (Blanding et al., 2006), requires the introduction of other assumptions/models in order to design an estimator that would ex-

tract from the true negatives new knowledge about possible values that the target state cannot take. In (Geller and Oshman, 2014), negative information is used to detect the absence of collision during the landing of a vehicle in order to improve its estimated state. (Allik, 2019) uses negative information to detect the absence of targets in an area monitored by UAVs in order to reduce the uncertainty regarding their estimated location. This is achieved during the update of the posterior probability. Nevertheless, unlike the authors of (Geller and Oshman, 2014) who tell that negative collision, *i.e.*, an absence of collision, can be obtained by using an accelerometer, (Allik, 2019) does not explain where the negative measurements come from. Only a target detection probability is introduced, as in (Dames, 2020; Hou et al., 2023; Li et al., 2024). Moreover, according to (Symington et al., 2010; Meera et al., 2019; Bai et al., 2021), the target detection probability is hard to quantify, especially in unknown environments.

2.4.5 . Idealized CVS

A target state can be composed of its location, its orientation, or its velocity. Regarding the target location, it may be defined as the coordinates of its center of mass, or its GPS location. Nonetheless, these components may be hard to estimate from a measurement provided by a camera. Some approaches such as (Mousavian et al., 2017) manage to solve this problem with 6D-pose detection algorithms, but they are not suitable for aerial drones whose computing power is inferior to that of ground drones. Other approaches, such as (Liu et al., 2022a), rely on the exploitation of a specific pixel, *e.g.*, the one located at the center of a bounding box provided by YOLO, associated to the detected target to evaluate a location that is considered representative of the actual target location. Thus, these approaches, designed for a pursuing problem where the target should always remains within the field of view of the drone, rather approximate the target location than solve an estimation problem. This may lead to errors that are hard to evaluate. (Goldhoorn et al., 2018; Kenmogne et al., 2019; Meera et al., 2019) have proposed different approaches to model the embedded CVS to solve a detection and localization problem. Nevertheless, most of the work addressing a target search and track problem frequently neglects this aspect of the problem by considering an idealized version of the embedded CVS (Kuhlman et al., 2017; Zhu et al., 2021; Zhao et al., 2022; Ibenthal et al., 2023; Zhang et al., 2024). Thus, this idealization often leads to assume that the robot gets directly a noisy measurement of the target state once located within the FoV.

Nowadays, very few approaches developed to solve either a CSAT or a SAR problem consider the difficulty of estimating the target location from CVS measurements. The more or less complex target shape, such as a human or a car, is no longer considered. Instead, the target shape is assumed punctual, see (Li et al., 2018; Dames, 2020; Ibenthal et al., 2021; Wei et al., 2024). It is detected only when its location enters the robot field of view and an estimation of its location - corrupted by some estimation noise - is directly obtained. This simplification has the advantage of proposing a framework allowing a simplified evaluation of a control design but at the cost of the introduction of assumptions regarding the capacity to collect and process new information, *e.g.*, a target detection probability or a model of the estimation uncertainty. According to (Bai et al., 2021) and (Symington et al., 2010), the choice to represent - most of the time - the localization uncertainty by a known noise and the capacity to detect the presence of a target with a probability of detection is not realistic. Therefore, the performance of the algorithms that use this simplified framework is likely to not be representative of the efficiency of

their approach once deployed in a real-life context with real CVS.

2.5 . Target search and tracking

The solution of a target search problem consists of three different steps : target search, target pursuit, and target tracking. The target search step consists of efficiently exploring the RoI to find targets that have not been previously identified. Once a mobile target is identified, a robot may be tasked to pursue the target and evolve within the RoI so as to keep the pursued target within range of its sensors. Additionally, unlike CMOMMT approaches, the robot may be tasked to return to a search mode and stop pursuing the target. Consequently, the robots must remain capable of recapturing these left-alone identified targets. For that purpose, the step of target tracking, *i.e.*, being able to track the target state even if it is not under observation, is to predict the future possible values of the target state by using previous knowledge of its estimated values with the knowledge the UAV has about the target dynamic.

Section 2.5.1 presents several approaches to model the presence/location of a target within the environment. Then, Section 2.5.2 provides some examples to search for a target. Section 2.5.3 explains how to pursue a target. Finally, Section 2.5.4 presents how a robot may remotely track a target.

2.5.1 . Representation of the target state

The first difficulty in exploring a RoI in order to find the targets is to define how the estimated target state is represented. In target localization problems, a common approach with stochastic assumptions is to rely either on a 2D (Yadav and Meng, 2014; Tang et al., 2019; Wei et al., 2024; Zhang et al., 2024; Li et al., 2024) or 3D (Li et al., 2018) discretization of the RoI in order to have access to a grid in which each cell stores the probability that it contains a target. This regular grid is often called a Target Probability Map (TPM), see Figure 2.14.

The resolution of the TPM is an issue since it has to cover the entirety of the RoI. A low resolution limits the accuracy of the estimated target location, but a high resolution leads to more storage and computation requirements. Alternative representations have been considered in (Vo et al., 2008; Dames, 2020; Banerjee and Schneider, 2024) where the authors model the set containing the location of all targets that are present within the RoI at some time instant by a Random Finite Set (RFS). It is a grid-free approach that relies on a set containing a finite and random number of random elements. As an example, one of the RFS introduced in (Dames, 2020) contains a random number of elements in which each element may be the value of the state vector related to a target. More specifically, the authors consider a Poisson RFS, which is defined as a set of independent and identically distributed elements where the cardinality of the set follows a Poisson distribution. One of the strengths of this approach is, since the number of elements to track is a variable, it is built to handle the disappearance of previously identified targets that have left the RoI as well as the appearance of new targets entering the RoI.

Moreover, as introduced in Section 2.4, set-membership approaches can be used if the measurement noise and other uncertainties are assumed bounded. Such approaches are defined by the characterization of a certain amount of possibly disconnected sets, their number being related to the number of identified targets. Each set estimate associated with an identified target contains the true

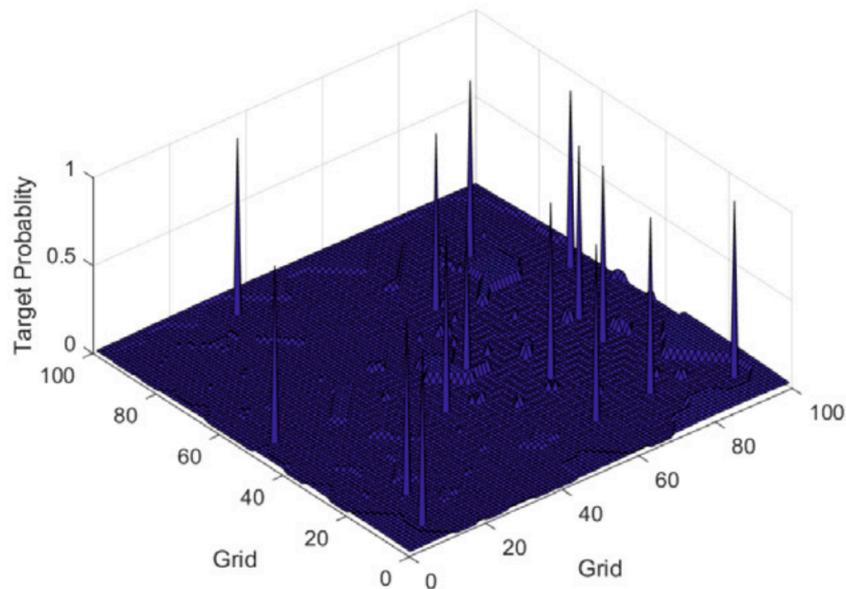


Figure 2.14 – Illustration of a 2D TPM which has been taken from the simulation results of (Li et al., 2024).

value of the target location. Such set can be characterized as a list of non-overlapping boxes (Jaulin, 2009; Drevelle and Bonnifait, 2011; Kenmogne et al., 2019), ellipsoids (Lu and Zhou, 2017; Hou et al., 2021), or polygons (Ibenthal et al., 2020). Figure 2.15 illustrates a polygon representation. These approaches can be used to determine an area of the RoI which may contain identified or unidentified targets, or areas that are guaranteed to contain no target location by using negative information. Such representation clearly defines areas to explore, to either find new targets, or capture previously identified targets.

2.5.2 . Target search

The most basic approach to explore an area of the RoI which may contain a target is to use a pre-planned trajectory (Niedzielski et al., 2021; Yanmaz, 2023). The authors in (Niedzielski et al., 2021) present an approach to explore an unknown cluttered environment to find one lost ill man. The man is able to move but, due to its illness, it will remain static at some point in time. They address the search aspect of the problem as follows. Since the target may be anywhere in the neighborhood of a Polish town, several RoIs are defined (using techniques detailed in the article). Once an area that may contain the target is defined, a UAV is tasked to explore the area, take photos, and upload the photos on a laptop for an automatic human detection which is corrected by a human operator. The exploration is performed using a lawnmower trajectory. Once the exploration of one of the RoIs is over, the UAV flies toward the next RoI to explore. Some advantages of their approach are (i) it is easy to deploy and it has been proved enough to save the man's life, and (ii) the pre-planned trajectory has been designed to reduce the impact of occlusions by unknown obstacles during the search of

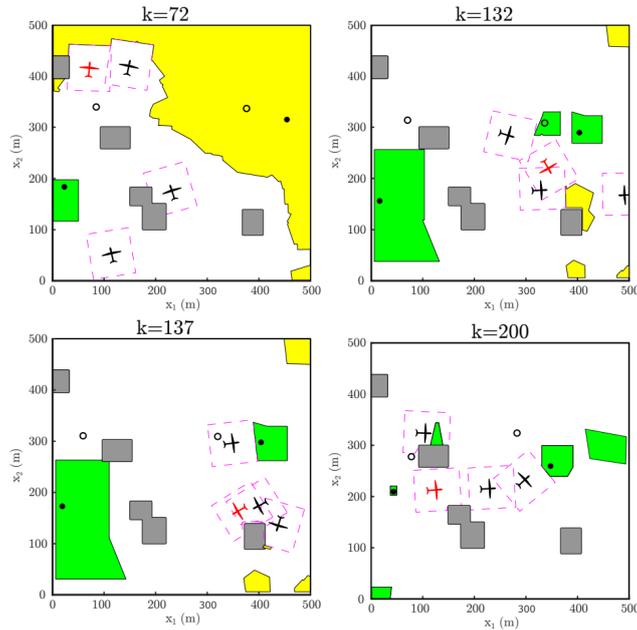


Figure 2.15 – Illustration from (Ibenthal et al., 2020) of the evolution at 4 different time instants of the possible locations of targets (black dots) that have been identified (green sets) and not-identified (yellow sets) by a fleet of 4 UAVs. The white circles are decoys that have not to be localized. The gray polygons represent the obstacles.

the ground target. However, such an approach has been designed for static targets. Therefore, with mobile targets that can enter an area already explored by the UAV, another search strategy has to be considered.

Unlike (Niedzielski et al., 2021), other approaches have been designed to actively account for the presence of obstacles during the target search. The method in (Goldhoorn et al., 2018) considers a ground robot. It exploits a known map to evaluate if the robot is able to observe a point or if it is hidden behind an obstacle, see equation (2.3). Then, during the target search, the robot searches for the cell that solves a trade-off between having a high probability to contain a target and having a fast reach time, *i.e.*, the path to reach the location must be short. Alternatively, the authors of (Meera et al., 2019) propose an approach to design a polynomial trajectory that drives the robot toward the next best observation point while avoiding obstacles. However, these approaches assume to have access to an accurate representation of the shape of the obstacles located within the RoI, which is not always ensured. To overcome this difficulty, mapping approaches may be used. The 3D cluttered environment in (Li et al., 2021) is unknown, but the robot is able to detect an obstacle and approximate its shape by a sphere. Then, the search problem is addressed with a combination of a TPM and a potential-field-based path-planning approach where discovered obstacles are used to generate repulsion fields in order to reach a desired location. The unknown environment in (Luo et al., 2023) is modeled using a 2D grid map. The uncertainty that a target is located within a given cell is modeled by a TPM. Additionally, some cells contain a static obstacle that the drones have to avoid. These cells are *a priori* unknown and detected only when the UAV is nearby. Once detected, they are avoided

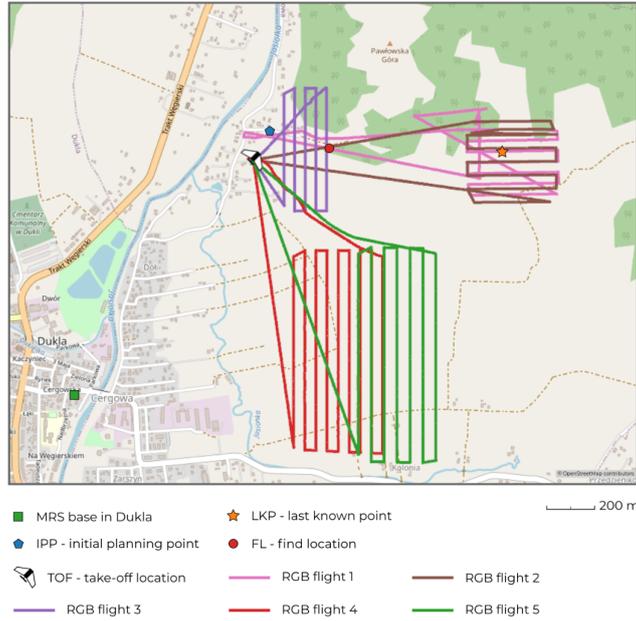


Figure 2.16 – Illustration of the searching strategy presented in (Niedzielski et al., 2021) where 4 RoIs (not represented) have been defined and explored using a lawnmower trajectory. The orange star and the blue pentagon are two locations that have been used to define the areas where the lost man may be. The red dot is the location where he has been found by the UAV.

during the path planning step which is based on the Dijkstra’s algorithm (these occupied cells have an infinite weight). The UAV achieves the target search by cooperatively evaluating the sequence of actions that maximizes the difference between the current uncertainty of the map and its predicted uncertainty according to a time horizon defined by an energy constraint. Targets are static and the uncertainty of a cell is reduced after having been searched by a UAV embedding a sensor modeled by a coefficient ϵ that represents its accuracy to detect a target, *i.e.*,

$$TPM_{k+1}(r, c) = (1 - \epsilon) TPM_k(r, c), \quad (2.5)$$

where r and c are the coordinates of the cell of the TPM that is updated.

(Kuhlman et al., 2017) consider a static target search in an environment with an uneven ground modeled by an elevation map, see Figure 2.17. A TPM is used with a Beta distribution to model that a target can appear and disappear from the robot FoV, *e.g.*, due to obstacle occlusion.

An alternative approach to handle the occlusions by unknown obstacles is to rely on a robot formation as in (Ibenthal et al., 2023). The authors assume a structured unknown environment. They assume that a single robot is unable to know if it has observed the ground where mobile targets are, or if it has observed an obstacle. To overcome this issue, they propose a geometric approach in order to evaluate an area on the ground that is proved to have been observed if the area is simultaneously monitored by several cooperative UAVs from a sufficient number of different points of view. This approach relies on the evaluation of a list of conic observation subsets used to define these different points of view. Then, if a point belonging to the ground has been observed by a UAV located in each

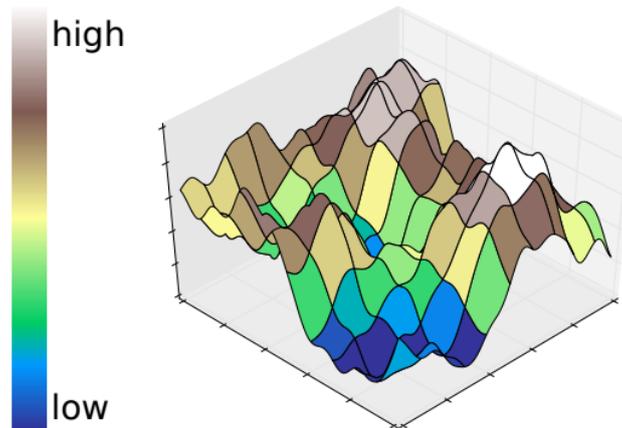


Figure 2.17 – Illustration from (Kuhlman et al., 2017) of an uneven ground generated by an elevation grid

of these conic observation subsets, and there is no detected target, then the point is proved to not be a target location. Using this approach, the target search is achieved by splitting the fleet of UAVs into groups and, then, controlling each group in order to maximize the size of the area that has been cooperatively observed. See Figure 2.18

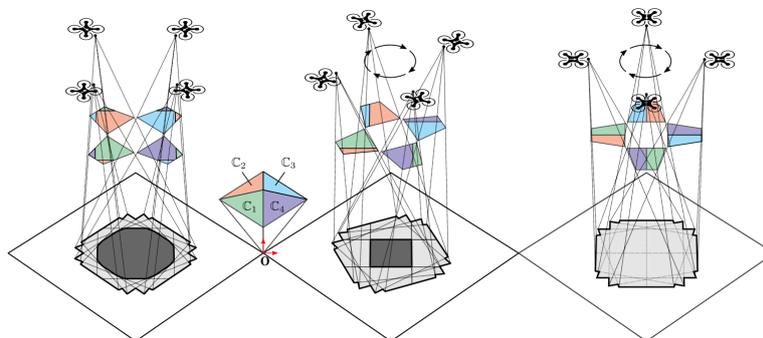


Figure 2.18 – Illustration taken from (Ibenthal et al., 2023) which represents the area theoretically covered by the 4 UAVs (light gray). A point on the ground is guaranteed to be observed only if it has been observed simultaneously by 4 UAVs where each one is located within one of the conic observation subsets represented by the geometrical observation conditions illustrated by the subset C_1 (green), C_2 (red), C_3 (blue), and C_4 (purple). The dark gray set is the area that has been proved to be observed even in the presence of unknown obstacles.

2.5.3 . Target pursuit

Once a target is identified and visible within the FoV of a robot, one can task the robot to simply pursue the target until an accurate estimation of its state is obtained. Assume that a bounding box in the image captured by its embedded camera is provided by a target detection algorithm such as

YOLO. Then a target pursuit may be achieved by controlling the robot to keep the bounding box close to the center of the image (Ribaric et al., 2004; Jin et al., 2010; Liu et al., 2022a). Such a technique may work in cluttered environments (Liu et al., 2018). Otherwise, (Ajmera and Singh, 2020) uses optical flow techniques to estimate the direction of motion of the pursued target from a sequence of images. The authors in (Silva et al., 2019) propose an approach of target re-identification based on the extraction of human features using a deep-learning algorithm. During the process, a bounding box is used to localize the target in the image. Then, to predict the future target location in the image, they assume that the target does not move significantly between two frames. Consequently, they dilate the initially obtained bounding box to get an area where the target would be. To pursue the target, they control the UAV to keep the target face in front of the camera.

2.5.4 . Target tracking

The strategy to track the state of a target without directly monitoring its evolution is strongly related to the representation of the estimated target state and must use some *a priori* knowledge. The tracking process usually involves a prediction-correction approach. Prior knowledge about the potential values of the tracked target state at a previous time instant, and the available information regarding the dynamics are used to predict its evolution at the current time. Then the prediction is corrected with the knowledge acquired from newly collected information (including negative information).

With a TPM, the prediction may be done by assuming that the target has a uniform probability to stay static in the previous cell it is located or to move to one of its neighboring cells (Wei et al., 2024). The predicted TPM is then evaluated as

$$\text{TPM}_{k+1|k}(r, c) = \frac{1}{9} \sum_{i=r-1}^{r+1} \sum_{j=c-1}^{c+1} \text{TPM}_k(i, j). \quad (2.6)$$

Then, the correction may be achieved using a Bayesian filter to evaluate the corrected TPM as

$$\text{TPM}_{k+1}(r, c) = \frac{\text{TPM}_{k+1|k}(r, c) \cdot p_{\text{TP}}}{\text{TPM}_{k+1|k}(r, c) \cdot p_{\text{TP}} + (1 - p_{\text{TP}}) (1 - \text{TPM}_{k+1|k}(r, c))}. \quad (2.7)$$

where p_{TP} is the probability of the UAV to truly detect a target in the cell.

Another approach to represent the probability of presence of a target is to use a particle filter as in (Wang et al., 2007; Goldhoorn et al., 2018; Xie et al., 2019). Knowing the target dynamic, one is able to evaluate the probability density function $p(\mathbf{x}_k^t | \mathbf{x}_{k-1}^t)$ of the current target state, knowing the previous state \mathbf{x}_{k-1}^t . Then, with all the measurements $\mathbf{y}_{1:k-1}$ that have been used since the beginning to evaluate the state of the target at time t_{k-1} , one is able to predict the probability density function of \mathbf{x}_k^t ,

$$p(\mathbf{x}_k^t | \mathbf{y}_{1:k-1}) = \int p(\mathbf{x}_k^t | \mathbf{x}_{k-1}^t) p(\mathbf{x}_{k-1}^t | \mathbf{y}_{1:k-1}) d\mathbf{x}_{k-1}^t. \quad (2.8)$$

The posterior probability of the actual target state, assuming the probability to get a measurement at time t_k knowing the target state is $p(\mathbf{y}_k | \mathbf{x}_k^t)$, is then

$$p(\mathbf{x}_k^t | \mathbf{y}_{1:k}) = \frac{p(\mathbf{x}_k^t | \mathbf{y}_{1:k-1}) p(\mathbf{y}_k | \mathbf{x}_k^t)}{p(\mathbf{y}_k | \mathbf{y}_{1:k-1})}. \quad (2.9)$$

However, the evaluation of such posterior probability may be difficult for nonlinear systems or non-Gaussian noise. To solve this issue, one can use the Monte Carlo Sampling by approximating the posterior distribution by N particles, *i.e.*, $p(\mathbf{x}_k^t | \mathbf{y}_{1:k}) \approx \sum_{i=1}^N \omega_k^{(i)} \delta(\mathbf{x}_k^t - \mathbf{x}_k^{t,(i)})$, where $\mathbf{x}_k^{t,(i)}$, for $i = 1, \dots, N$, is a particle and $\omega_k^{(i)}$ is the weight associated to the particle i . The weight represents the confidence that the particle $\mathbf{x}_k^{t,(i)}$ is the target state. The particles are obtained using importance sampling which relies on the introduction of an easy-to-sample distribution q . The weights are obtained using the update rule

$$\omega_k^{(i)} = \omega_{k-1}^{(i)} \frac{p(\mathbf{y}_k | \mathbf{x}_k^{t,(i)}) p(\mathbf{x}_k^{t,(i)} | \mathbf{x}_{k-1}^{t,(i)})}{q(\mathbf{x}_k^{t,(i)} | \mathbf{x}_{k-1}^{t,(i)}, \mathbf{y}_k)}. \quad (2.10)$$

The tracking is achieved by predicting the possible new values of the target state using the particles, followed by a step of resampling.

Another problem arises regarding the tracking of multiple targets with a Bayes filter. To address a multi-target tracking problem, two RFSs can be used to model a discrete set of targets and measurements. However, the use of RFSs with classic Bayes filters leads to a high computational burden due to the evaluation of the multiple integrals that have to be considered (Vo and Ma, 2006; Tang et al., 2015). To avoid this problem, the first statistical moment λ of the probability distribution of the RFS related to the targets, also named the Probability Hypothesis Density (PHD) or the intensity, is considered. The integral of the PHD over the region of interest gives the expected cardinality of the target-RFS. Poisson RFS holds a particular place within this approach since it is entirely defined by its first moment. Then, with this approach and under other assumptions that are detailed in (Vo and Ma, 2006), the PHD filter, which tracks the evolution of the PHD, is

$$\lambda_{k|k-1}(\mathbf{x}_k^t) = b(\mathbf{x}_k^t) + \int_{\text{RoI}} d(\mathbf{x}_{k-1}^t) p(\mathbf{x}_k^t | \mathbf{x}_{k-1}^t) \lambda_{k-1}(\mathbf{x}_{k-1}^t) d\mathbf{x}_{k-1}^t \quad (2.11)$$

$$\lambda_k(\mathbf{x}_k^t) = (1 - p(\mathbf{x}_k^t | \mathbf{x}_k^u)) \lambda_{k|k-1}(\mathbf{x}_k^t) \quad (2.12)$$

$$+ \sum_{\mathbf{y} \in \mathcal{Y}_k} \frac{\psi_{\mathbf{y}, \mathbf{x}_k^u}(\mathbf{x}_k^t) \lambda_{k|k-1}(\mathbf{x}_k^t)}{p_{\text{FP}}(\mathbf{y} | \mathbf{x}_k^u) + \int_{\text{RoI}} \psi_{\mathbf{y}, \mathbf{x}_k^u}(\mathbf{x}) \lambda_{k-1}(\mathbf{x}) d\mathbf{x}} \quad (2.13)$$

where $b()$ and $d()$ are the intensities of birth (a target entering the RoI) and survival (the target staying in the RoI). $p(\mathbf{x}_k^t | \mathbf{x}_k^u)$ is the detection probability knowing that the robot is located at \mathbf{x}_k^u . $\psi_{\mathbf{y}, \mathbf{x}_k^u}(\mathbf{x}_k^t)$ is the probability that the robot at location \mathbf{x}_k^u would get the measurement \mathbf{y}_k if the target is at \mathbf{x}_k^t . $p_{\text{FP}}(\mathbf{y} | \mathbf{x}_k^u)$ is the probability of getting a false positive, *i.e.*, getting a measurement of a target state that is not present. This filter has been used in (Dames, 2020; Banerjee and Schneider, 2024).

A target tracking algorithm based on a set-membership approach also involves a prediction-correction mechanism. Consider that $\mathbb{X}_{j,k}^t$ represents the set estimate containing the possible locations of target j at time t_k , and $\overline{\mathbb{X}}_k^t$ is the set estimate gathering all the possible locations of still unidentified targets. Consider that the target dynamics is $\mathbf{x}_{k+1}^t = f(\mathbf{x}_k^t, \mathbf{w}_k)$ where $\mathbf{w}_k \in \mathbb{W}$ is an unknown but bounded state perturbation. The prediction step consists in characterizing the possible future locations of the tracked target $\mathbb{X}_{j,k+1|k}^t$ and the still unidentified targets $\overline{\mathbb{X}}_{k+1|k}^t$, knowing $\mathbb{X}_{j,k}^t$ and $\overline{\mathbb{X}}_k^t$. Equation (2.14)

defines how to characterize those sets, with \mathbb{X}_k and $\mathbb{X}_{k+1|k}$ instead of $\mathbb{X}_{j,k}^t$ and $\mathbb{X}_{j,k+1|k}^t$, or $\overline{\mathbb{X}}_k^t$ and $\overline{\mathbb{X}}_{k+1|k}^t$.

$$\mathbb{X}_{k+1|k} = \{ \mathbf{x} \mid \exists \mathbf{p} \in \mathbb{X}_k, \exists \mathbf{w} \in \mathbb{W}, \mathbf{x} = f(\mathbf{p}, \mathbf{w}) \}. \quad (2.14)$$

Then, using a measurement obtained at time t_{k+1} , (Ibenthal et al., 2023) characterizes two sets. With negative information, they characterize a set $\mathbb{X}_{k+1}^{t,neg}$ gathering all the current locations where no target can be. If target j has been identified at time t_{k+1} , then the UAV gets a set estimate $\mathbb{X}_{j,k+1}^{t,m}$ containing all the locations of target j which are consistent with the noisy measurements. Finally, the correction step consists of an intersection or a difference, *i.e.*,

$$\mathbb{X}_{j,k+1}^t = \left(\mathbb{X}_{j,k+1|k}^t \cap \mathbb{X}_{j,k+1}^{t,m} \right) \setminus \mathbb{X}_{k+1}^{t,neg}. \quad (2.15)$$

If target j is a newly detected target, *i.e.*, $\mathbb{X}_{j,k+1|k}^t$ does not exist, then we rather have

$$\mathbb{X}_{j,k+1}^t = \left(\overline{\mathbb{X}}_{k+1|k}^t \cap \mathbb{X}_{j,k+1}^{t,m} \right) \setminus \mathbb{X}_{k+1}^{t,neg}. \quad (2.16)$$

Finally, the correction of $\overline{\mathbb{X}}_{k+1|k}^t$ is simply

$$\overline{\mathbb{X}}_{k+1}^t = \overline{\mathbb{X}}_{k+1|k}^t \setminus \mathbb{X}_{k+1}^{t,neg}. \quad (2.17)$$

Note that if target j has not been identified at time t_{k+1} , then $\mathbb{X}_{j,k+1}^{t,m}$ does not exist and should be removed from equation (2.15).

However, the intersection may be difficult to characterize. Therefore, an inner or outer approximation can be computed with algorithms such as the Set Inverter Via Interval Analysis (SIVIA) (Jaulin and Walter, 1993).

2.6 . Control design

Once the information collected by the sensors has been processed by the embedded CVS and an estimator has been used to get new knowledge about the presence and location of obstacles or targets, the robot has to update its current control input according to a pre-defined strategy. Section 2.6.1 introduces the different approaches available to coordinate the robots, with their advantages and drawbacks. Then, Section 2.6.2 presents some examples of control design before presenting in Section 2.6.3 the model predictive control.

2.6.1 . Coordination between the robots

There exist three different approaches to address the problem of coordination between the robots in a team : the centralized approach, the decentralized approach, and the distributed approach.

The centralized approach assumes that all the robots are able to communicate with a central node. This central node receives from the robots new measurements/knowledge that is processed to evaluate the next best control input for each robot. This list of control laws is defined as the course of actions of the fleet that is obtained, *e.g.*, by solving an optimization problem reflecting the objectives to achieve. The central node has access to all the knowledge of the fleet, and usually has enough

computational power to design the control law for each robot. Nevertheless, the measurements have to be gathered at the central node, which may be difficult when the RoI is large, usually due to limited communication range constraints. Moreover, since each robot relies on the central node, if the central node fails, the whole fleet is disrupted.

The decentralized approach assumes a heterogeneous fleet of robots where some of them act as mobile local nodes, *i.e.*, a robot that has access to enough computational power to design the control input for any robot that is connected to it. Therefore, the fleet is partitioned into groups of robots. Each group contains one of these mobile local nodes, *i.e.*, a leader, whereas the others, *i.e.*, the followers, are only here to collect new information and stay close enough to the associated leader. Consequently, unlike the centralized approach, an advantage of the decentralized approach is that it solves the problem of limited communication range. However, if the leader has a problem, all of its followers are disabled. Moreover, since the leader has access, most of the time, only to the measurements of its followers, the evaluated control laws may be suboptimal.

The distributed approach assumes that each robot has the computing power to evaluate its own control input by only accounting for the knowledge it has access to and the knowledge broadcast by its neighbors. Since each robot is assumed autonomous, the lack of communication between robots is no longer a problem. However, if a robot is isolated from the others, then, even if it is able to update its control input, it cannot cooperate with the others. Thus the evaluated control laws may be suboptimal as well.

2.6.2 . Examples of control design approaches

CSAT problems aim to navigate in a RoI to search and identify all targets and maintain an accurate estimation of their state. Consequently, the control strategies that have to be deployed to address these problems to guide the robots toward positions where they would cooperatively maximize the information they collect. See (Lyu et al., 2023) for a survey of SAR algorithms and the considered strategies.

In the presence of known obstacles, (Meera et al., 2019) iteratively updates the UAV trajectory by designing a new polynomial trajectory passing through $N - 1$ waypoints. At each iteration, a list of possible future UAV locations is generated by a random sampling of locations in the environment. To avoid obstacles, each new sampled location must be visible from the previous location. Then, the possible future location which maximizes a criterion that considers the information collected, the risk of collision, and the flight time is taken as the next waypoint for the updated polynomial trajectory. The authors evaluate the information the UAV would collect by comparing the trace of the (prior) covariance matrix of the Gaussian process with the one obtained (*i.e.*, the posterior covariance matrix) at the expected location. During the planning, the UAV is assumed to detect no target.

In (Goldhoorn et al., 2018), a ground robot either pursues a target or enters in a search mode if it has lost it. One of their proposed tracking approaches relies on a particle filter. The search mode evaluates the list of cells containing the highest density of particles. Then, knowing the map and the obstacles, it searches for the best cell to reach. The path planning is done with a combination of a Dijkstra algorithm (for known static obstacles) and a trajectory roll-out planner (for unknown dynamic obstacles).

The trajectory design for target search and obstacle avoidance in (Li et al., 2021; Zhen et al., 2020) is

done with attractive and repulsive fields. Both use a TPM. (Zhen et al., 2020) directly use a gradient of the TPM to evaluate its attraction force, whereas (Li et al., 2021) proceeds to a nonlinear transformation of the TPM in order to get a map of uncertainty regarding the presence of a target in each cell, see Figure 2.19. Then, using this evaluated uncertainty, (Li et al., 2021) define a criterion that compares the gain in going toward an attractive location with the gain in going toward unexplored cells.

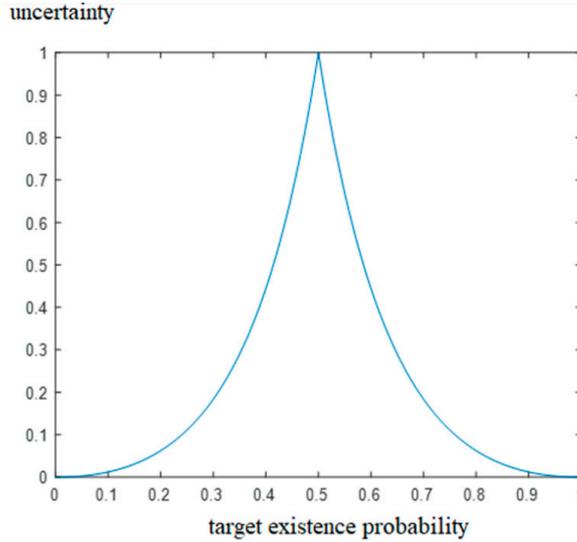


Figure 2.19 – Figure from (Li et al., 2021). The nonlinear transformation between the probability that a target exists in a given cell, *i.e.*, $\text{TPM}_k(r, c)$, and what they defined as the uncertainty of the cell, *i.e.*, $\mu_k(r, c) = \exp(-k |\ln(1/\text{TPM}_k(r, c)) - 1|)$.

(Dames, 2020) uses a PHD filter with the Lloyd’s algorithm to guide the robots toward areas with a high density of targets. The Lloyd’s algorithm is defined as the minimization of

$$J(\{\mathbf{x}_{i,k}^u\}_{i \in \{1, \dots, N^u\}}) = \int_{\text{RoI}} \min_{i \in \{1, \dots, N^u\}} (d(\mathbf{x}, \mathbf{x}_{i,k}^u))^2 \lambda_k(\mathbf{x}) d\mathbf{x} \quad (2.18)$$

and is used to update a Voronoi partition. N^u is the number of robots, and $d(\mathbf{x}, \mathbf{y})$ evaluates the distance between \mathbf{x} and \mathbf{y} . A Voronoi cell is associated to each robot, and the barycenter of the Voronoi cells is the location the robot has to reach. A similar partition-based approach is used in (Sun and Tsiotras, 2017; Dang et al., 2019).

When the target is cooperative, (Tang et al., 2019) proposes an approach combining particle swarm optimization with a fruit-fly optimization algorithm to search and reach a target in an unknown cluttered environment. The fruit-fly optimization algorithm (Pan, 2012) is an algorithm where the target (the fruit) emits some signal with an intensity that decreases with the distance to the robots (the fruit-flies). Other pheromone-based algorithms exist, *e.g.*, (Cheng et al., 2024; Li et al., 2024), but they are rather used to enhance the area coverage rather than designed for the search of mobile targets.

Another possible control design is based on the partially observable Markov decision process (POMDP), see (Astrom, 1965; Lauri et al., 2022) and the references therein. A POMDP is a stochastic path-planning approach relying on the modeling of a decision process of the robot, also denoted as

the agent. The environment and what is contained in it are represented with a finite set of discrete states the agent can have, actions the agent can take, and observations the agent can acquire. The goal is to find the best sequence of actions that optimizes a criterion. (Zhu et al., 2021) considers communication problems and uncertain UAV location, and (Zhang et al., 2024) adds a modified multi-agent reinforcement learning layer to enhance the cooperative behavior of the robots.

Alternatively, the authors of (Wei et al., 2024) propose a control design that searches the trajectory which minimizes the probability of not detecting a target. For that purpose, knowing that the probability of one robot to detect a target is p_{TP} , they introduce the probability that the fleet fails to detect a target with $p_{fail} = \prod_{i=1}^{N^u} (1 - p_{TP})$, N^u being the number of robots within the fleet. From that probability, they define a criterion with some constraints related to the robot dynamics. Then, to solve their proposed optimization problem, the authors use deep reinforcement learning (a combination of Multi-Agent Reinforcement Learning and Proximal Policy Optimization) to search for the best sequence of control inputs in order to find the targets.

Multi-agent reinforcement learning approaches (Hou et al., 2023; Zhang et al., 2024) aim to emulate an interaction between the robot and the environment in order to learn a strategy that optimizes a criterion. Nonetheless, as pointed out in (Li et al., 2024), multi-agent reinforcement learning approaches are not considered suitable to solve a target search and tracking problem because the data base used for the training may be not representative of the actual situation. Similarly, strategies based on Markov decision processes are not scalable to large and complex outdoor environments.

Potential-field-based strategies, as in (Zhen et al., 2020; Li et al., 2021), enable a natural evaluation of the direction to follow in order to search for targets while avoiding obstacles. Otherwise, control designs based on the update of the current long-term trajectory, as in (Meera et al., 2019; Luo et al., 2023), search for the best new trajectory that optimizes a criterion. The criterion is generally based on the maximization of the expected gain of information, *i.e.*, information that can be processed to reduce the uncertainty regarding the estimated state of the targets. Moreover, the criterion can be completed with other cost functions for obstacle avoidance or incorporate other approaches such as a pheromone-based methods, see (Li et al., 2024).

2.6.3 . Predictive control

Unlike classical controllers, such as the PID controller, predictive controllers rely on a model of the system to predict its future behavior when a sequence of control inputs $\mathbf{u}_{k:h} = \{\mathbf{u}_k, \dots, \mathbf{u}_{k+h-1}\}$ is applied over a control horizon composed of h steps (Christofides et al., 2013; Schwenger et al., 2021). This prediction allows to evaluate the distance between the desired reference state and the expected state of the system over a prediction horizon. Then, the effectiveness of the sequence $\mathbf{u}_{k:h}$ is evaluated by using a criterion J accounting for that distance and possibly other terms such as the energy consumption, and the smoothness of the trajectory. Predictive controllers aim to search for the sequence $\hat{\mathbf{u}}_{k:h}$ which optimizes J .

According to (Schwenger et al., 2021) predictive controllers are easier to implement than classical controllers. The reason is classical controllers involve the definition of explicit control laws which would become more tedious or even impossible for constrained nonlinear systems. On the other hand, predictive controllers mostly rely on a model of the system, that is often well-defined and easily accessible, and on a criterion J to determine its desired behavior. The price to pay is an increased

computational burden since the predictive controller has to update at each time step the sequence of control inputs.

The application of Model Predictive Control (MPC) for target search and tracking is slightly different from what is presented in (Schwenzer et al., 2021). Regarding robot i , the MPC still searches for the sequence of control inputs $\hat{\mathbf{u}}_{i,k:h}$ which optimizes a criterion J . However, instead of evaluating the deviation between a reference and future measured state of the system, it evaluates the expected capacity of the robot to reduce the uncertainty of the estimated target states.

This section presents the MPC by comparing the MPC used in (Li et al., 2024) (stochastic approach) and in (Ibenthal et al., 2021) (set-membership approach) to solve a CSAT problem.

At time t_k , the state of robot i is $\mathbf{x}_{i,k}^u$, with $i \in \{1, \dots, N^u\}$ where N^u is the number of robots. To search for the optimal sequence of control inputs $\hat{\mathbf{u}}_{i,k:h}$, the MPC considers a candidate sequence of control inputs $\mathbf{u}_{i,k:h} \in \mathbb{U} \times \dots \times \mathbb{U}$, where \mathbb{U} is a known set of feasible control inputs, evaluates its impact on the reduction of uncertainty, and reiterates until convergence. For instance, both (Li et al., 2024) and (Ibenthal et al., 2021) consider a UAV flying at a fixed altitude and speed and the set of feasible control inputs is a finite set of heading (yaw) angle increments. In this case, the number of possible candidate sequences of control inputs is finite. Since the dynamics of the UAV is perfectly known, a sequence of states $\mathbf{x}_{i,k:h|k}^u = \{\mathbf{x}_{i,k+1|k}^u, \dots, \mathbf{x}_{i,k+h|k}^u\}$ is deduced. If the robot dynamics is uncertain or is affected by some external disturbances, tube MPC can be used (Le et al., 2011; Cannon et al., 2012; Lopez et al., 2019).

To evaluate the impact of a candidate sequence of control inputs $\mathbf{u}_{i,k:h}$, one has to evaluate the expected gain of knowledge the robot would acquire at each of its future states, *i.e.*, $\mathbf{x}_{i,k:h|k}^u$. In (Li et al., 2024), who use a TPM, the authors identify which cells would be under observation and, if the probability of target presence is above 0.8, then they assume to detect a target. Then, they evaluate the reduction of uncertainty for each cell within sensing range by computing the difference of uncertainty between two time instants, where the uncertainty is defined as

$$\xi_k(r, c) = -\text{TPM}_{k-1}(r, c) \log_2(\text{TPM}_{k-1}(r, c)) \quad (2.19)$$

$$- (1 - \text{TPM}_{k-1}(r, c)) \log_2(1 - \text{TPM}_{k-1}(r, c)), \quad (2.20)$$

For a set-membership approach, the reduction of uncertainty is represented by a reduction of the size of the set estimates after each new observation. Nonetheless, it is often not possible to ensure that a target will be observed in the future. Therefore, the simplifying assumption that no target will be observed is considered, as in (Ibenthal et al., 2021). Consequently, with the robot i FoV $\mathbb{F}(\mathbf{x}_{i,k+\ell|k})$, where $\ell = 1, \dots, h$, and without obstacles, they update the set estimates by only removing potential locations that would have been observed, *i.e.*,

$$\mathbb{X}_{j,k+\ell}^{\text{t,P}} = \mathbb{X}_{j,k+\ell|k}^{\text{t,P}} \setminus \mathbb{F}(\mathbf{x}_{i,k+\ell|k}) \quad (2.21)$$

where $\mathbb{X}_{j,k+\ell}^{\text{t,P}}$ is the correction of the predicted set estimate $\mathbb{X}_{j,k+\ell|k}^{\text{t,P}}$ containing the potential locations at time $t_{k+\ell}$ of the previously identified target j , knowing its set of estimated locations at time t_k , *i.e.*, $\mathbb{X}_{j,k}^{\text{t,P}}$. The characterization of $\mathbb{X}_{j,k+\ell|k}^{\text{t,P}}$ follows a similar set-membership prediction process described by (2.14), *i.e.*,

$$\mathbb{X}_{j,k+\ell|k}^{\text{t,P}} = \left\{ f(\mathbf{p}, \mathbf{w}) \mid \mathbf{p} \in \mathbb{X}_{j,k+\ell-1}^{\text{t,P}}, \mathbf{w} \in \mathbb{W} \right\}, \quad (2.22)$$

with the difference that if $\ell = 1$, then $\mathbb{X}_{j,k}^{t,P} = \mathbb{X}_{j,k}^t$. The same approach is applied regarding the set estimate which may still contain unidentified targets $\bar{\mathbb{X}}_k^t$.

The criterion to evaluate the impact of $\mathbf{u}_{i,k:h}$ may be rather simple as in (Ibenthal et al., 2021) where the impact of $\mathbf{u}_{i,k:h}$ is evaluated by computing the area of what remains to explore, *i.e.*,

$$J(\mathbf{u}_{i,k:h}) = \phi \left(\bar{\mathbb{X}}_{k+h}^{t,P} \cup \bigcup_{j \in \mathcal{L}_{i,k}} \mathbb{X}_{j,k+h}^{t,P} \right) \quad (2.23)$$

$$+ \lambda d \left(\mathbf{x}_{i,k+h|k}, \bar{\mathbb{X}}_{k+h}^{t,P} \cup \bigcup_{j \in \mathcal{L}_{i,k}} \mathbb{X}_{j,k+h}^{t,P} \right) \quad (2.24)$$

where $\phi(\mathbb{X})$ is the area of $\mathbb{X} \subset \mathbb{R}^2$, and $\mathcal{L}_{i,k}$ is the list known to UAV i that contains the indices of all the targets that have been identified at time t_k . The second part of (2.23) evaluates the closest distance between the set estimates and the UAV location.

The criterion in (Li et al., 2024) is

$$J(\mathbf{u}_{i,k:h}) = \lambda_1 R_{\text{targ}}^{i,k} + \lambda_2 R_{\text{uncertainty}}^{i,k} + \lambda_3 R_{\text{phero}}^{i,k} + R_{\text{par}}^{i,k} + R_{\text{collision}}^{i,k} \quad (2.25)$$

where $R_{\text{targ}}^{i,k}$ evaluates the number of targets that would be detected, according to the assumption that a target would be detected if its associated probability of presence is above 0.8. $R_{\text{uncertainty}}^{i,k}$ evaluates the uncertainty reduction. $R_{\text{phero}}^{i,k}$ is a pheromone-based cost function to encourage the UAV to explore unexplored areas. $R_{\text{par}}^{i,k}$ is to keep the UAV within its dedicated area to explore. $R_{\text{collision}}^{i,k}$ is to avoid collisions.

Depending on the context, other criteria may be defined, as in (Fei et al., 2022) where the problem of limited communication range is addressed by introducing a component in the cost function to keep neighbors close enough to maintain communication, but spread enough to maintain a large covering area.

For cooperative design, each robot should account for the sequence of control inputs of the other robots in order to move toward a location that still solves the target search and track problem while being beneficial for the rest of the fleet. For a fleet composed of N^u robots, the following joint optimization problem can be considered

$$\{\hat{\mathbf{u}}_{i,k:h}\}_{i \in \mathcal{N}^u} = \arg \min_{\{\mathbf{u}_{i,k:h}\}_{i \in \mathcal{N}^u}} \sum_{i \in \mathcal{N}^u} J(\mathbf{u}_{i,k:h}). \quad (2.26)$$

Nonetheless, the resolution of such optimization problems may be difficult and time-consuming. Therefore, alternative approaches and simplifications have to be considered. For example, \mathbb{U} may be finite. Otherwise, the authors of (Ibenthal et al., 2021) consider a suboptimal cooperative approach where each UAV computes its optimal sequence of control inputs only once it has received the optimal sequence from all of the other UAVs within communication range and with a smaller index. This allows a simpler update of the trajectory while maintaining some cooperative behavior between the drones since, except for the first one, each UAV accounts for the trajectory of its predecessors to update its own.

2.7 . Conclusion of the chapter

This chapter has presented an overview of some approaches to address the main subproblems that compose a CSAT problem. The acquisition of new measurements from the embedded CVS and the exploitation of these measurements to solve a target localization problem. Different approaches have been presented to represent the potential locations of both unidentified and identified targets. Then, for each representation some approaches to either explore the RoI to search for the target, or to remotely track their potential location, are presented. Target pursuing approaches have also been discussed as well as approaches to map the RoI or to control the robots.

Clearly, the CSAT problem is a complex problem to address. Simplifying assumptions are often made, such as the idealization of the CVS and the target location estimator. Those assumptions have the advantage of simplifying the evaluation of a proposed control design. Unfortunately, the performance of the proposed control design is strongly related to the reliability of those simplifying assumptions that are unjustifiable in most cases. The reason is, CVS measurements and, more importantly, measurements related to the detection and the identification of a target, are obtained by deep learning algorithms. The acquisition of these measurements is conditioned by the training data set that has been used and the observation conditions - which involve the weather conditions and the presence of obstacles -. Therefore, the approximation of the process to detect a target by a probability of detection is difficult to justify, even more if unknown obstacles are located within the RoI. Moreover, since the performance of a target location estimator is strongly related to the CVS measurements that are used as inputs, the idealization of the estimator with a known estimation noise is also hard to justify.

The main objective of this thesis is to propose a new approach to take into account the complex measurements provided by the CVS to maintain the most accurate estimation of the target locations. Depth-maps, classified images, and bounding boxes related to the detection and identification of a target are considered. A set-membership estimator is developed to exploit these CVS measurements in order to characterize set estimates that either contain the true location of a target or are proved to contain no target location. Then, the tracking algorithm of (benthal, 2022) is adapted to update the knowledge of a UAV regarding the potential location of both identified and unidentified targets.

Unlike CMOMMT approaches, CSAT approaches aim to determine the exact number of targets inside an RoI. Therefore, until the fleet of UAVs determines that there cannot be more unidentified targets within the RoI, the UAVs have to simultaneously track all already identified targets while exploring the RoI to find new ones. Assuming the targets outnumber the UAVs, a trade-off has to be solved between searching for new targets and pursuing identified targets to keep an accurate estimation of their locations. To solve this trade of, a MPC is used and a new criterion is defined to account for the presence of unknown 3D obstacles.

The motivation behind the design of this new criterion is that there exists few CSAT algorithms that consider an unknown and cluttered 3D environment with a mapping approach. On the one hand, the absence of a 3D mapping approach can be justified since a UAV has a limited storage capacity and computing power. On the other hand, new and more storage-efficient mapping approaches have been presented for urban environments. Thus, by using CVS measurements to map efficiently the unknown environment, a new criterion inspired by (Meera et al., 2019) is proposed to improve the target search within a cluttered RoI.

Chapter 3 - Hypotheses and Problem formulation

Consider an unknown number N^t of targets spread within a Region of Interest (RoI) containing an unknown number N^o of structured obstacles of unknown location. A fleet of N^u cooperative Unmanned Aerial Vehicles (UAVs) is deployed to find and identify these targets and maintain an accurate estimation of their location. This Cooperative Search, Acquisition, and Tracking (CSAT) problem is formalized in this chapter.

The assumptions and models to represent the structured environment, the targets, and the UAVs are detailed. Each UAV embeds a Computer Vision System (CVS) consisting of a camera and computer vision algorithms to get measurements from the environment. These algorithms process images collected by the camera. There is, among them, an image segmentation, a depth map evaluation, and a target detection/identification algorithm. These algorithms are instrumentals in getting measurements related to the presence of an identified target or to assess its absence.

We choose to address the CSAT problem with a set-membership approach. The approach assumes that the measurement noises and other uncertainties are bounded with known bounds. Thus, we can design a set-membership estimator that characterizes sets that are guaranteed to contain the location of the targets, provided that the hypotheses on the measurement models and noise bounds are satisfied. Few previous methods developed to address a CSAT problem exploit CVS measurements in a set-membership approach. This is mainly because obtaining measurement models for a CVS involving deep learning algorithms is difficult. Thus, the measurement error is hard to characterize. To address this issue, this chapter introduces assumptions and models to relate the CVS measurements with the targets and obstacles present in the RoI. With these assumptions and models, the CVS measurements can be exploited by a set-membership estimator to characterize these set estimates that either are guaranteed to contain the location of a target or, with negative information, are guaranteed to not contain a target location.

The assumptions introduced here are used to design a new target location estimator in Chapter 4 and a control approach for the UAVs in Chapter 5. The estimator exploits CVS measurements related to targets and obstacles to update the knowledge the UAV has regarding the potential locations of both identified and unidentified targets. The proposed estimator is also used to update the knowledge of the UAV regarding the location and shape of the observed obstacles. Then, a control algorithm uses this updated knowledge to update the trajectory of the UAV.

3.1 . Environment model

Consider an environment with a frame \mathcal{F} attached. In what follows, several other frames are introduced, such as the frames attached to the UAVs. To simplify the notations, vectors with no superscript related to a frame are implicitly expressed in the reference frame \mathcal{F} .

The Region of Interest (RoI) \mathbb{X}_0 is a bounded subset of the 3D space, *i.e.*, $\mathbb{X}_0 \subset \mathbb{R}^2 \times \mathbb{R}^+$, containing the targets, the UAVs, and the obstacles. The bounds of the RoI are assumed to be fixed and known to the UAVs. The ground of the RoI is assumed to be flat and horizontal, *i.e.*, $\mathbb{X}_g = \{\mathbf{x} \in \mathbb{X}_0 \mid x_3 = 0\}$,

where x_3 is the third coordinates of a vector \mathbf{x} , representing the height. The ground of RoI may contain several elements such as roads, sidewalks, or soils, but, their height is considered negligible. In addition, their particular nature is assumed irrelevant to the target search. Therefore, they are merged in \mathbb{X}_g .

N^o fixed obstacles are spread within the RoI. They are assumed to be indexed with indexes in the set $\mathcal{N}^o = \{1, \dots, N^o\}$. The shape of the m -th obstacle, *i.e.*, the part of the RoI occupied by the m -th obstacle, is denoted $\mathbb{S}_m^o \subset \mathbb{R}^3$. \mathbb{S}_m^o is assumed to be Z -convex, *i.e.*,

$$\forall \mathbf{x} \in \mathbb{S}_m^o, \forall \lambda \in [0, 1], \lambda \mathbf{x} + (1 - \lambda) \mathbf{p}_g(\mathbf{x}) \in \mathbb{S}_m^o, \quad (3.1)$$

where $\mathbf{p}_g(\mathbf{x})$ is the projection of \mathbf{x} on the ground \mathbb{X}_g . Figure 3.1 illustrates this assumption.

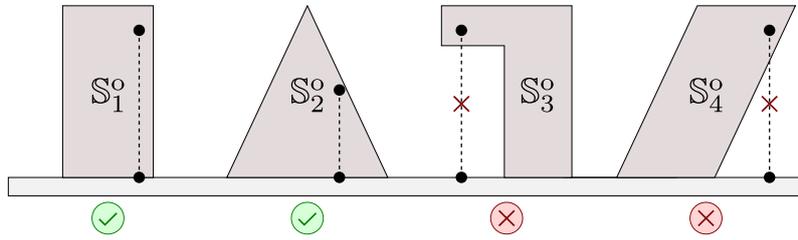


Figure 3.1 – Illustration of different obstacles that may or may not satisfy equation (3.1).

This assumption is relevant for most of the buildings, as pointed out by (Zhang et al., 2023). Elements of the environment such as trees or streetlights may be present but do not satisfy (3.1) and, thus no such element is considered present in the environment, see Chapter 8 for an extension. The footprint of obstacle $m \in \mathcal{N}^o$ is the subset of \mathbb{X}_g that belongs to \mathbb{S}_m^o , *i.e.*, $\mathbb{S}_m^o \cap \mathbb{X}_g$. According to 3.1, the footprint of an obstacle is also $\mathbf{p}_g(\mathbb{S}_m^o)$.

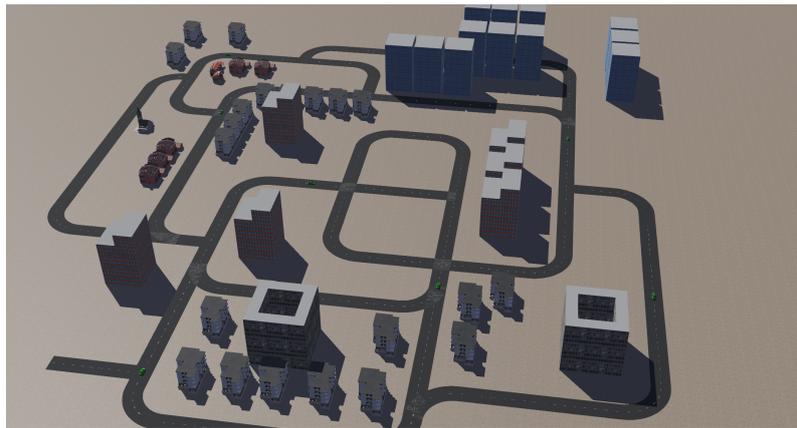


Figure 3.2 – Illustration of the urban RoI simulated on Webots (Webots,) used to evaluate the performance of our approach in Chapter 5. Only buildings satisfying (3.1) are considered.

3.2 . Target model

Let N^t be the unknown but fixed number of ground targets evolving in the RoI \mathbb{X}_0 . Targets are uniquely identified with indexes in the set $\mathcal{N}^t = \{1, \dots, N^t\}$. All targets are assumed to be vehicles of the same type, and they do not enter or leave \mathbb{X}_0 .

The state vector of target $j \in \mathcal{N}^t$ at time t_k is $\mathbf{x}_{j,k}^t$. This state vector contains, among others, the vector $\mathbf{x}_{j,k}^t$ of the coordinates of the center of gravity of target j . In this thesis, we only consider ground targets evolving on \mathbb{X}_g . Moreover, we limit the target search to a 2D space. Thus, we consider that the location of target j at time t_k is the projection of $\mathbf{x}_{j,k}^t$ on \mathbb{X}_g , denoted as $\mathbf{x}_{j,k}^{tg} = \mathbf{p}_g \left(\mathbf{x}_{j,k}^t \right)$, rather than $\mathbf{x}_{j,k}^t$. We do not try to estimate its orientation.

We consider that the behavior of target j can be modeled by

$$\mathbf{x}_{j,k+1}^{tg} = \mathbf{f}^t \left(\mathbf{x}_{j,k}^{tg}, \mathbf{v}_{j,k}^t \right), \quad (3.2)$$

where \mathbf{f}_j^t is known and $\mathbf{v}_{j,k}^t$ represents the unknown but bounded target control input as well as the state perturbations at time k . The vector $\mathbf{v}_{j,k}^t$ is only assumed to belong to a known box $[\mathbf{v}^t]$.

The space occupied by target j , *i.e.*, its shape, is the subset $\mathbb{S}^t \left(\mathbf{x}_{j,k}^t \right) \subset \mathbb{R}^2 \times \mathbb{R}^+$ and depends on $\mathbf{x}_{j,k}^t$. Since the targets are cars, the target shape is rigid and $\mathbb{S}^t \left(\mathbf{x}_{j,k}^t \right)$ depends mainly on the target position and orientation. If the shape evolves, other components of $\mathbf{x}_{j,k}^t$ must be considered. We also assume that the center of gravity of a target is always located inside its shape, *i.e.*,

$$\forall j \in \mathcal{N}^t, \mathbf{x}_{j,k}^t \in \mathbb{S}^t \left(\mathbf{x}_{j,k}^t \right). \quad (3.3)$$

The target location $\mathbf{x}_{j,k}^{tg}$, however, does not necessarily belong to $\mathbb{S}^t \left(\mathbf{x}_{j,k}^t \right)$, as for cars where $\mathbf{x}_{j,k}^{tg}$ is located between the wheels. The shape of the tracked targets is unknown to the UAVs, but their type (car, truck. . .) is usually known. Consequently, some information about the target dimensions is available.

Assumption 1. *We assume that a vertical circular right cylinder $\mathbb{C}^t \left(\mathbf{x}_{j,k}^{tg} \right)$ of known height h^t and radius r^t with basis centered in $\mathbf{x}_{j,k}^{tg}$ can be defined such that, for any target state $\mathbf{x}_{j,k}^t$*

$$\mathbb{S}^t \left(\mathbf{x}_{j,k}^t \right) \subset \mathbb{C}^t \left(\mathbf{x}_{j,k}^{tg} \right). \quad (3.4)$$

Figure 3.3 illustrates Assumption 1. The projection $\mathbf{p}_g \left(\mathbb{C}^t \left(\mathbf{x}_{j,k}^{tg} \right) \right)$ of $\mathbb{C}^t \left(\mathbf{x}_{j,k}^{tg} \right)$ on \mathbb{X}_g is the disc of center $\mathbf{x}_{j,k}^{tg}$ and radius r^t .

3.2.1 . Interaction of a target with its environment

This section introduces some assumptions about the interaction of a target and its environment, including other targets.

The r -neighborhood of a set $\mathbb{S} \subset \mathbb{X}_0$ is defined as

$$\mathbb{N}(\mathbb{S}, r) = \{ \mathbf{x} \in \mathbb{X}_0 \mid d(\mathbf{x}, \mathbb{S}) \leq r \} \quad (3.5)$$

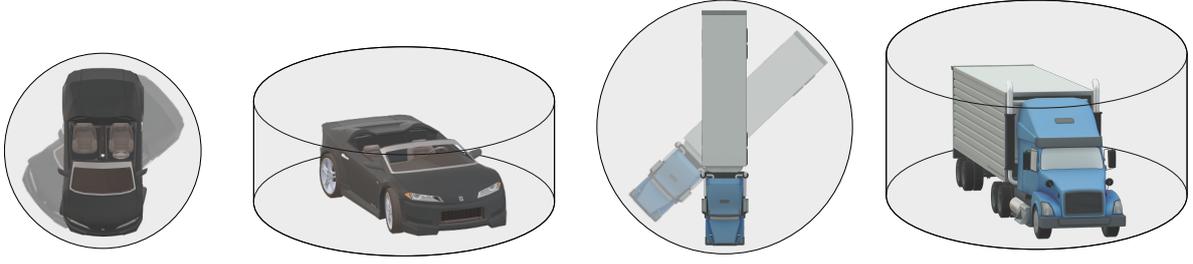


Figure 3.3 – Illustration of a possible vertical circular right cylinder $\mathbb{C}^t(\mathbf{x}_{j,k}^{tg})$ containing the shape $\mathbb{S}^t(\mathbf{x}_{j,k}^t)$ of a car-like or truck-like target

with $d(\mathbf{x}, \mathbb{S}) = \min_{\mathbf{y} \in \mathbb{S}} \|\mathbf{x} - \mathbf{y}\|$.

We also introduce the r -ground neighborhood of a set $\mathbb{S} \subset \mathbb{X}_g$

$$\mathbb{N}_g(\mathbb{S}, r) = \{\mathbf{x} \in \mathbb{X}_g \mid d(\mathbf{x}, \mathbb{S}) \leq r\}. \quad (3.6)$$

One can remark that $\mathbb{N}_g(\{\mathbf{x}\}, r)$ is a disc of center $\mathbf{x} \in \mathbb{X}_g$, and radius r . Thus, another way to express $\mathbf{p}_g(\mathbb{C}^t(\mathbf{x}_{j,k}^{tg}))$ is $\mathbb{N}_g(\{\mathbf{x}_{j,k}^{tg}\}, r^t)$.

With Obstacles

We assume that a target cannot enter any obstacle. Therefore,

$$\forall j \in \mathcal{N}^t, \forall m \in \mathcal{N}^o, \mathbb{S}^t(\mathbf{x}_{j,k}^t) \cap \mathbb{S}_m^o = \emptyset. \quad (3.7)$$

As the shape of targets is unknown, (3.7) is difficult to exploit. Besides, we aim to only determine the target location $\mathbf{x}_{j,k}^{tg}$. Thus, we assume that the distance between $\mathbf{x}_{j,k}^{tg}$ and any obstacle is always larger than some known safety distance r_s^{to} , i.e.,

$$\forall j \in \mathcal{N}^t, \forall m \in \mathcal{N}^o, \mathbb{N}_g(\{\mathbf{x}_{j,k}^{tg}\}, r_s^{to}) \cap \mathbf{p}_g(\mathbb{S}_m^o) = \emptyset \quad (3.8)$$

$$\Leftrightarrow \mathbf{x}_{j,k}^{tg} \notin \mathbb{N}_g(\mathbf{p}_g(\mathbb{S}_m^o), r_s^{to}). \quad (3.9)$$

Figure 3.4 illustrates (3.8).

Remark 2. We assumed that the target shape is always contained in the cylinder \mathbb{C}^t of radius r^t , then a possible choice for r_s^{to} is $r_s^{to} = r^t$.

With other targets

We assume also that two targets never collide, i.e.,

$$\forall j \in \mathcal{N}^t, \forall \ell \in \mathcal{N}^t \setminus \{j\}, \mathbb{S}^t(\mathbf{x}_{j,k}^t) \cap \mathbb{S}^t(\mathbf{x}_{\ell,k}^t) = \emptyset. \quad (3.10)$$

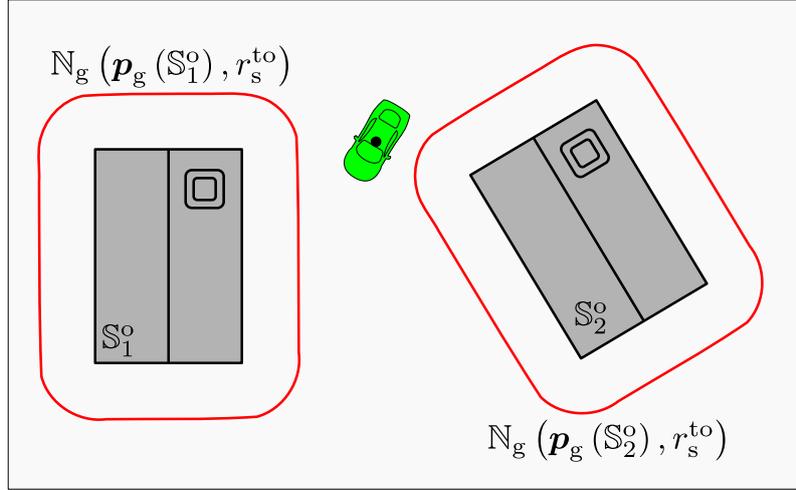


Figure 3.4 – Illustration of the r_s^{to} -ground neighborhood (in red) of two obstacles (houses, in gray). Target is in green and its location is the black dot.

However, as in Section 3.2.1, this assumption is difficult to exploit. Alternatively, one considers that the distance between the location of target j and any other target location is always larger than r_s^{tt} , i.e.,

$$\forall j \in \mathcal{N}^t, \forall \ell \in \mathcal{N}^t \setminus \{j\}, \mathbf{x}_{\ell,k}^{tg} \notin \mathbb{N}_g \left(\left\{ \mathbf{x}_{j,k}^{tg} \right\}, r_s^{tt} \right). \quad (3.11)$$

Figure 3.5 illustrates (3.11).

Remark 3. Another formulation for (3.11) is

$$\forall j \in \mathcal{N}^t, \forall \ell \in \mathcal{N}^t \setminus \{j\}, \mathbb{N}_g \left(\left\{ \mathbf{x}_{j,k}^{tg} \right\}, r_s^{tt}/2 \right) \cap \mathbb{N}_g \left(\left\{ \mathbf{x}_{\ell,k}^{tg} \right\}, r_s^{tt}/2 \right) = \emptyset. \quad (3.12)$$

Thus, since we assumed that the target shape is always contained in the cylinder \mathbb{C}^t of radius r^t , then a possible choice for r_s^{tt} is to take $r_s^{tt} = 2r^t$.

3.3 . UAV model

A fleet of N^u cooperative UAVs is deployed within \mathbb{X}_0 to search, find and track the targets. UAVs are identified with indexes in the set $\mathcal{N}^u = \{1, \dots, N^u\}$. Each UAV embeds a CVS, including a camera, described and modeled as reported below.

At time t_k , the state vector $\mathbf{x}_{i,k}^u$ of UAV $i \in \mathcal{N}^u$ contains, among others, the location of its center of gravity $\mathbf{x}_{i,k}^u \in \mathbb{R}^2 \times \mathbb{R}^{+*}$ and its attitude $\Theta_{i,k}^u \in \mathbb{R}^3$. The vector $\mathbf{x}_{i,k}^u$ evolves as

$$\mathbf{x}_{i,k+1}^u = \mathbf{f}^u \left(\mathbf{x}_{i,k}^u, \mathbf{u}_{i,k}^u, \mathbf{w}_{i,k}^u \right). \quad (3.13)$$

The function \mathbf{f}^u is known; $\mathbf{u}_{i,k}^u$ represents the control input, assumed to belong to a finite set of feasible control inputs \mathbb{U} ; $\mathbf{w}_{i,k}^u$ represents some unknown perturbation only assumed to be contained in the known box $[\mathbf{w}^u] = [\underline{\mathbf{w}}^u, \overline{\mathbf{w}}^u]$. Thus, if $\mathbf{w}_{i,k}^u \neq 0$ at some time instant, the state vector of UAV i

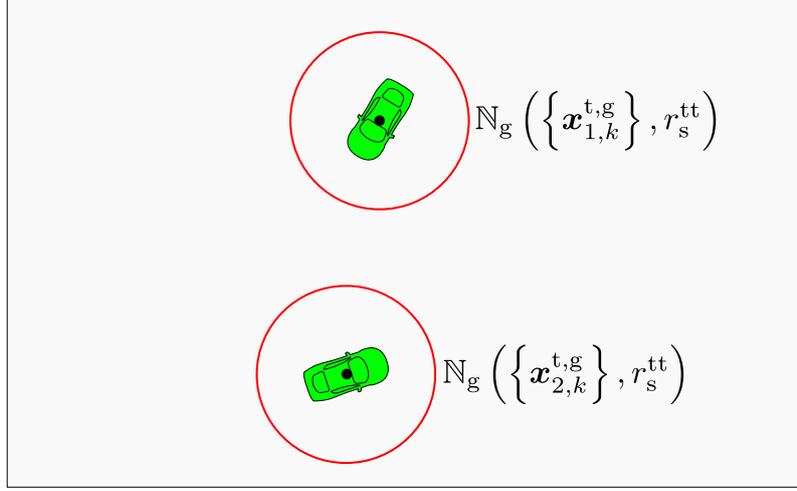


Figure 3.5 – Illustration of (3.11). The red sets are the boundaries of the r_s^{tt} -ground neighborhoods of the two target locations (black dots).

becomes uncertain. In Chapters 4 and 5, the UAVs are assumed to perfectly know their state vector, at each time instant. It is only in Chapter 6 that we consider an uncertain UAV dynamics and, thus, have to account for the state uncertainty of the UAV in the estimation process.

Each UAV embeds proprioceptive sensors such as Inertial Measurement Units (IMU) or GPS receivers to measure its state. The measurement $\mathbf{y}_{i,k}^u$ of the state of UAV i by those sensors is assumed to be described as

$$\mathbf{y}_{i,k}^u = \mathbf{h}^u(\mathbf{x}_{i,k}^u) + \mathbf{w}_{i,k}^{u,m}, \quad (3.14)$$

where $\mathbf{w}_{i,k}^{u,m}$ is some measurement noise, unknown but bounded in the known box $[\mathbf{w}^{u,m}] = [\underline{\mathbf{w}}^{u,m}, \overline{\mathbf{w}}^{u,m}]$.

The shape of UAV i is $\mathbb{S}^u(\mathbf{x}_{i,k}^u)$ and is perfectly known. A body frame $\mathcal{F}_{i,k}^b$ with origin $\mathbf{x}_{i,k}^u$ is attached to UAV i . The rotation matrix from $\mathcal{F}_{i,k}^b$ to \mathcal{F} is denoted $\mathbf{M}_{\mathcal{F}_{i,k}^b}^{\mathcal{F}}$. $\mathbf{M}_{\mathcal{F}_{i,k}^b}^{\mathcal{F}}$ depends on $\Theta_{i,k}^u$. To simplify the notation, the time index is omitted for the frames attached to a UAV.

The coordinates of some vector $\mathbf{x} \in \mathbb{R}^3$, when expressed in \mathcal{F}_i^b , are

$$\mathbf{T}_{\mathcal{F}_i^b}^{\mathcal{F}}(\mathbf{x}) = \mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}}(\mathbf{x} - \mathbf{x}_{i,k}^u). \quad (3.15)$$

3.3.1. Communication

Data exchange within the fleet is modeled by the undirected graph $\mathcal{G}_k = (\mathcal{N}^u, \mathcal{E}_k)$, with \mathcal{N}^u the set of vertices and $\mathcal{E}_k \subset \mathcal{N}^u \times \mathcal{N}^u$ the set of edges of the graph. \mathcal{E}_k describes the connectivity at time t_k . For two UAVs $i \in \mathcal{N}^u$ and $i' \in \mathcal{N}^u$, if $(i, i') \in \mathcal{E}_k$, then both UAVs are able to exchange information without delay and error. The set $\mathcal{N}_{i,k}^u = \{i' \in \mathcal{N}^u \mid (i, i') \in \mathcal{E}_k\}$ contains all UAV indexes to which UAV i is able to communicate at time t_k . For simplicity, we consider that $i \in \mathcal{N}_{i,k}^u$.

For more details regarding the various communication protocols that may be used for UAVs, see (Shi et al., 2021).

3.4 . Model of the camera and CVS

This section presents a geometric approach to model the way UAVs perceive their environment and collect information that will be processed by their CVS, see also Section 3.5.

Section 3.4.1 introduces a geometric model for the embedded camera that is used to get images of the monitored environment. Section 3.4.2 uses this model to characterize the FoV of the camera, and Section 3.4.3 presents an approach to deduce from this geometric model the coordinates on the CCD array of any point belonging to the environment.

To simplify the notations in this section, the time index is omitted.

3.4.1 . Camera model

Each UAV of the fleet embeds the same type of camera. An image \mathbf{I}_i provided by the camera of UAV i consists of N_r rows and N_c columns.

A camera frame \mathcal{F}_i^c is attached to the camera of UAV i . This frame has its origin located at the camera optical center $\mathbf{x}_i^{c, \mathcal{F}_i^b}$, and is oriented such that the positive z -axis represents the optical axis of the camera, the x -axis is parallel to the pixel rows in the CCD array and the y -axis is parallel to the columns. The rotation matrix from \mathcal{F}_i^b to \mathcal{F}_i^c is noted $\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}_i^c}$.

Usually, the camera's intrinsic parameters such as the dimensions of the CCD cells or the focal length are known. Table 3.1 gathers all the camera intrinsic parameters that are used in this work.

Notation	Relation	Definition (unit)
f	-	camera focal length (m)
α_c/α_r	-	camera horizontal/vertical aperture (rad)
H_c/H_r	$H_* = 2f \tan\left(\frac{\alpha_*}{2}\right)$	dimensions of the CCD array (m)
f_c/f_r	$f_* = f N_*/H_*$	camera focal length (pixels)

Table 3.1 – Camera's intrinsic parameters. 'c' is for column, horizontal, or x -axis; 'r' is for row, vertical, or y -axis

We consider the pinhole model without distortion presented by (Faugeras, 1993). With this model, the matrix of intrinsic parameters

$$\mathbf{K} = \begin{pmatrix} -f_c & 0 & N_c/2 \\ 0 & -f_r & N_r/2 \end{pmatrix}. \quad (3.16)$$

is used to define the projection $\mathbf{p}_{\mathcal{F}_i^c}(\cdot)$ of some point $\mathbf{x}^{\mathcal{F}_i^c} \in \mathbb{R}^3$ belonging to the environment onto the CCD array in order to obtain the 2D coordinates on the CCD array of its image, *i.e.*,

$$\mathbf{p}_{\mathcal{F}_i^c}(\mathbf{x}^{\mathcal{F}_i^c}) = \mathbf{K} \mathbf{x}^{\mathcal{F}_i^c} / x_3^{\mathcal{F}_i^c}, \quad (3.17)$$

where $x_3^{\mathcal{F}_i^c}$ is the third coordinate of $\mathbf{x}^{\mathcal{F}_i^c}$. See Figure 3.6 for an illustration of the pinhole model.

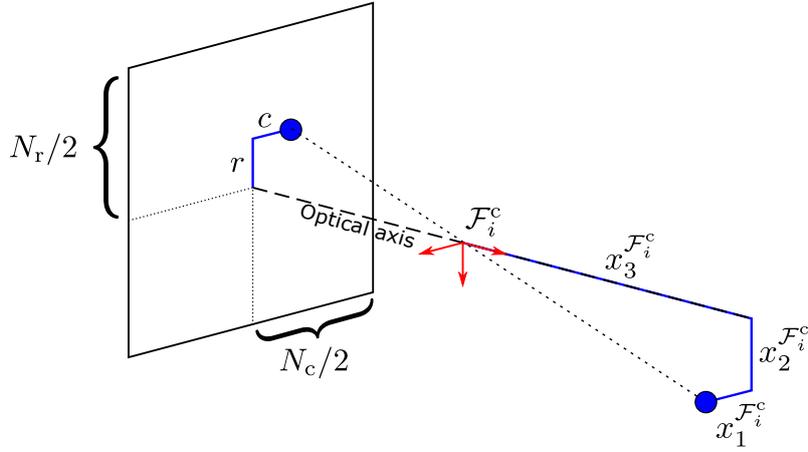


Figure 3.6 – Illustration of the pinhole model : $c = -f_c x_1^{F_i^c} / x_3^{F_i^c} + N_c/2$ and $r = -f_r x_2^{F_i^c} / x_3^{F_i^c} + N_r/2$.

3.4.2 . Field of view

According to the geometric approach (3.17) to model the camera, each light ray passing through the optical center of the camera and illuminating the CCD array at $(c, r) \in [0, N_c] \times [0, N_r]$ can be modeled by a half-line of direction

$$\mathbf{v}^{\mathcal{F}_i^c}(c, r) = \frac{1}{\nu(c, r)} \begin{pmatrix} (\frac{N_c}{2} - c) / f_c \\ (\frac{N_r}{2} - r) / f_r \\ 1 \end{pmatrix}, \quad (3.18)$$

where

$$\nu(c, r) = \sqrt{((\frac{N_c}{2} - c) / f_c)^2 + ((\frac{N_r}{2} - r) / f_r)^2 + 1} \quad (3.19)$$

is a normalization coefficient. See Figure 3.7 for an illustration of some light rays illuminating a pixel (n_r, n_c) . The set

$$\mathcal{V}_i(n_r, n_c) = \left\{ \mathbf{M}_{\mathcal{F}_i^c}^{\mathcal{F}} \mathbf{v}^{\mathcal{F}_i^c}(c, r) \mid c \in [n_c - 1, n_c], r \in [n_r - 1, n_r] \right\} \quad (3.20)$$

gathers the directions, expressed in \mathcal{F} , of all light rays contributing to the illumination of the pixel (n_r, n_c) . The matrix $\mathbf{M}_{\mathcal{F}_i^c}^{\mathcal{F}} = \mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}} \mathbf{M}_{\mathcal{F}_i^c}^{\mathcal{F}_i^b}$ describes the rotation from \mathcal{F}_i^c to \mathcal{F} .

From this geometric representation of the camera, one can represent the FoV of the camera of UAV i (dubbed as FoV of UAV i in what follows) with the set

$$\mathbb{F}(\mathbf{x}_i^u) = \left\{ \mathbf{x}_i^c + \sum_{\ell=1}^4 a_\ell \mathbf{M}_{\mathcal{F}_i^c}^{\mathcal{F}} \mathbf{v}_\ell^{\mathcal{F}_i^c} \mid a_\ell \in \mathbb{R}^+ \right\}, \quad (3.21)$$

with

$$\mathbf{v}_\ell^{\mathcal{F}_i^c} = \mathbf{v}^{\mathcal{F}_i^c}(c_\ell, r_\ell), \ell = 1, \dots, 4, \quad (3.22)$$

where

$$(c_\ell, r_\ell) \in \left\{ (N_c, 0), (0, 0), (N_c, N_r), (0, N_r) \right\}. \quad (3.23)$$

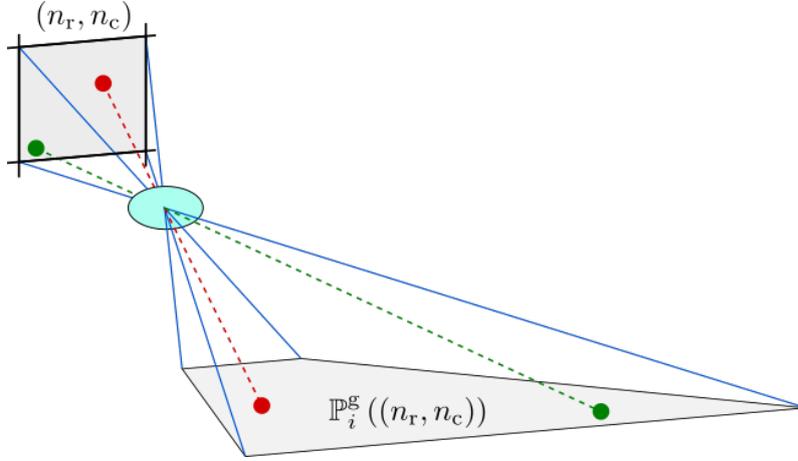


Figure 3.7 – Some light rays illuminate of the pixel (n_r, n_c) ; The 4 blue lines are the light rays illuminating the corners of (n_r, n_c) ; The quadrangle $\mathbb{P}_i^g((n_r, n_c))$ is defined in Section 4.3.1.

The FoV of UAV i denotes the set of points in \mathbb{R}^3 that are potentially observed in the RoI by a UAV when its state is \mathbf{x}_i^u (ignoring obstacles and targets).

In addition, one assumes that the CVS cannot reliably process information related to parts of the FoV situated too far away from the optical center of the camera. On the contrary, any information related to some $\mathbf{x} \in \mathbb{F}(\mathbf{x}_i^u)$ at a distance from \mathbf{x}_i^c smaller than d_{\max} can be reliably processed by the CVS, where d_{\max} is some a known distance. Therefore, the set of points of the FoV that can be reliably processed by the CVS embedded on UAV i is

$$\underline{\mathbb{F}}(\mathbf{x}_i^u) = \mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{B}(\mathbf{x}_i^c, d_{\max}), \quad (3.24)$$

where $\mathbb{B}(\mathbf{x}_i^c, d_{\max})$ is the ball of \mathbb{R}^3 of center \mathbf{x}_i^c and radius d_{\max} .

3.4.3 . Projection on camera frame

This section introduces the transform to obtain the 2D coordinates in the CCD array of any point expressed in the reference frame.

First, consider the transform $\mathbf{T}_{\mathcal{F}^b}^{\mathcal{F}^c}(\cdot)$ expressing in \mathcal{F}_i^c the coordinates of a point $\mathbf{x}^{\mathcal{F}^b}$ (expressed in the body frame of UAV i)

$$\mathbf{T}_{\mathcal{F}^b}^{\mathcal{F}^c}(\mathbf{x}^{\mathcal{F}^b}) = \mathbf{M}_{\mathcal{F}^b}^{\mathcal{F}^c}(\mathbf{x}^{\mathcal{F}^b} - \mathbf{x}_i^{c, \mathcal{F}^b}). \quad (3.25)$$

Composing (3.15) and (3.25), one gets the transform

$$\mathbf{T}_{\mathcal{F}}^{\mathcal{F}^c} = \mathbf{T}_{\mathcal{F}^b}^{\mathcal{F}^c} \circ \mathbf{T}_{\mathcal{F}}^{\mathcal{F}^b} \quad (3.26)$$

to express in \mathcal{F}_i^c the coordinates of \mathbf{x} , initially expressed in \mathcal{F} .

By combining (3.17) with (3.26), one obtains the function $\mathbf{p}_c(\mathbf{x}_i^u, \mathbf{x})$ which provides the 2D coordinates in the CCD array of UAV i of any point \mathbf{x} of the environment

$$\mathbf{p}_c(\mathbf{x}_i^u, \mathbf{x}) = \mathbf{p}_{\mathcal{F}_i^c}(\mathbf{T}_{\mathcal{F}}^{\mathcal{F}^c}(\mathbf{x})). \quad (3.27)$$

For any pixel $(n_r, n_c) \in \mathcal{N}^l$, the notation $\mathbf{p}_c(\mathbf{x}_i^u, \mathbf{x}) \in (n_r, n_c)$ indicates that the projection of \mathbf{x} on the CCD array belongs to that pixel, *i.e.*,

$$\mathbf{p}_c(\mathbf{x}_i^u, \mathbf{x}) \in ([n_c - 1, n_c], [n_r - 1, n_r])^T. \quad (3.28)$$

Moreover, for any $\mathbf{x} \in \mathbb{F}(\mathbf{x}_i^u)$, there exists a pixel $(n_r, n_c) \in \mathcal{N}^l$ such that there exists a vector $\mathbf{v} \in \mathcal{V}_i(n_r, n_c)$ that is colinear with the line $(\mathbf{x}_i^c, \mathbf{x})$. Consequently, since \mathbf{x} may have contributed to the illumination of pixel (n_r, n_c) , then the projection of \mathbf{x} on the CCD must belong to $([n_c - 1, n_c], [n_r - 1, n_r])^T$, *i.e.*,

$$\mathbf{x} \in \mathbb{F}(\mathbf{x}_i^u) \Rightarrow \exists (n_r, n_c) \in \mathcal{N}^l, \mathbf{p}_c(\mathbf{x}_i^u, \mathbf{x}) \in (n_r, n_c). \quad (3.29)$$

3.5 . Information provided by the CVS

At time t_k , the camera generates the image $\mathbf{I}_{i,k}$. It is fed into the CVS which provides a depth-map $\mathbf{D}_{i,k}$, obtained using, *e.g.*, (Park and Kim, 2012; Madhuanand et al., 2021; Shimada et al., 2023). An array of pixel labels $\mathbf{L}_{i,k}$ is also obtained using, *e.g.*, (Grilli et al., 2017; Howard et al., 2019), as well as a list $\mathcal{D}_{i,k}^t$ of identified targets. For each of these targets, a box $[\mathcal{Y}_{i,j,k}^t]$ in the image $\mathbf{I}_{i,k}$ containing pixels of the identified targets $j \in \mathcal{D}_{i,k}^t$ may be obtained using, *e.g.*, (Redmon et al., 2016; Minaeian et al., 2018; Luo et al., 2022; Jiang et al., 2022; Liu et al., 2022a). We assume that the depth map $\mathbf{D}_{i,k}$ and the array of labels $\mathbf{L}_{i,k}$ have the same size as $\mathbf{I}_{i,k}$. Besides, at time t_k , the RoI, consisting of the ground \mathbb{X}_g , the obstacles \mathbb{S}_m^o , $m \in \mathcal{N}^o$, the targets $\mathbb{S}^t(\mathbf{x}_{j,k}^t)$, $j \in \mathcal{N}^t$, and the UAVs $\mathbb{S}^u(\mathbf{x}_{i,k}^u)$, $i \in \mathcal{N}^u$, can be represented as

$$\mathbb{S}_k^{\text{RoI}} = \mathbb{X}_g \cup \bigcup_{m \in \mathcal{N}^o} \mathbb{S}_m^o \cup \bigcup_{j \in \mathcal{N}^t} \mathbb{S}^t(\mathbf{x}_{j,k}^t) \cup \bigcup_{i \in \mathcal{N}^u} \mathbb{S}^u(\mathbf{x}_{i,k}^u). \quad (3.30)$$

UAV i is only able to collect information from the portion of the RoI that is located within its FoV, *i.e.*, $\mathbb{F}(\mathbf{x}_{i,k}^u) \cap \mathbb{S}_k^{\text{RoI}}$. Information related to obstacles, targets, or parts of the ground in $\mathbb{F}(\mathbf{x}_{i,k}^u) \setminus \mathbb{F}(\mathbf{x}_{i,k}^u)$ may be reliable, but is not considered as such in what follows.

To keep the notation simple, the time index is again omitted in this section.

In the following sections, assumptions and models are introduced to describe the way measurements provided by the CVS can be related to the environment (ground, obstacles, targets, and other UAVs) of each UAV. These models will be instrumental in the estimation algorithms described in Chapter 4.

3.5.1 . Depth map

The CVS provides an array of $N_c \times N_r$ depth measurements \mathbf{D}_i . Each element (n_r, n_c) of \mathbf{D}_i is assumed to be an estimation of the distance between the optical center \mathbf{x}_i^c of the camera of UAV i and the part of the environment that has reflected light rays illuminating the pixel (n_r, n_c) .

Consider a pixel $(n_r, n_c) \in \mathcal{N}^l$. The distance $\rho(\mathbf{x}_i^c, \mathbf{v})$ between the optical center of the camera \mathbf{x}_i^c

and the environment along a vector $\mathbf{v} \in \mathcal{V}_i(n_r, n_c)$ is

$$\rho(\mathbf{x}_i^c, \mathbf{v}) = \min \left\{ d_{\mathbf{v}}(\mathbf{x}_i^c, \mathbb{X}_g), d_{\mathbf{v}}\left(\mathbf{x}_i^c, \bigcup_{m \in \mathcal{N}^o} \mathbb{S}_m^o\right), \right. \quad (3.31)$$

$$\left. d_{\mathbf{v}}\left(\mathbf{x}_i^c, \bigcup_{j \in \mathcal{N}^t} \mathbb{S}^t(\mathbf{x}_j^t)\right), d_{\mathbf{v}}\left(\mathbf{x}_i^c, \bigcup_{\ell \in \mathcal{N}^u} \mathbb{S}^u(\mathbf{x}_\ell^u)\right) \right\} \quad (3.32)$$

where $d_{\mathbf{v}}(\mathbf{x}, \mathbb{S})$ is the distance from a point $\mathbf{x} \in \mathbb{X}_0$ to the intersection between the set \mathbb{S} and the half-line of origin \mathbf{x} and direction \mathbf{v} . If there is no intersection, then $d_{\mathbf{v}}(\mathbf{x}, \mathbb{S})$ is infinite.

We assume that the measured distance $\mathbf{D}_i(n_r, n_c)$ associated to pixel (n_r, n_c) is such that

$$\exists \mathbf{v} \in \mathcal{V}_i(n_r, n_c), \mathbf{D}_i(n_r, n_c) = \rho(\mathbf{x}_i^c, \mathbf{v})(1 + w), \quad (3.33)$$

where w is some unknown noise assumed to belong to the known interval $[\underline{w}, \bar{w}]$. In what follows,

$$\mathbf{D}_i^0(n_r, n_c) = \frac{\mathbf{D}_i(n_r, n_c)}{1 + w} \quad (3.34)$$

is used to represent the noise-free distance, i.e., $\rho(\mathbf{x}_i^c, \mathbf{v})$, that would be measured with $w = 0$.

As $w \in [\underline{w}, \bar{w}]$, the interval

$$[\mathbf{D}_i](n_r, n_c) = \left[\frac{1}{1 + \bar{w}}, \frac{1}{1 + \underline{w}} \right] \mathbf{D}_i(n_r, n_c), \quad (3.35)$$

contains $\mathbf{D}_i^0(n_r, n_c)$.

3.5.2 . Pixel classifier

The CVS provides an array of $N_c \times N_r$ pixel labels \mathbf{L}_i . In this work, only four classes of pixel labels are considered, namely *Ground*, *Target*, *Obstacle*, and *Unknown*. We assume that all pixels labeled by the CVS as *Ground*, *Target*, or *Obstacle* are reliably labeled. For example, pixels with *Ground* labels actually correspond to elements of the RoI such as roads, soils, or sidewalks. When the *Target* label is used for some pixels, we assume that they actually correspond to the observation of, at least, parts of a target. When a pixel is labeled *Obstacle*, light rays reflected by an obstacle have actually contributed to the illumination of that pixel. Pixels labeled *Unknown* correspond to other UAVs, or to elements of the environment that are too far to be reliably identified by the CVS.

The estimation approaches considered in this work are much more sensitive to misclassified pixels than to pixels without labels. Usually, the confidence threshold of pixel classifiers may be adjusted to get a compromise between the proportion of misclassified pixels and not labeled pixels (Jiang et al., 2023). Here, the threshold should be tuned in such a way to avoid misclassified pixels.

Considering the four classes, four sets of pixel indexes are deduced from \mathbf{L}_i , namely \mathcal{Y}_i^g , \mathcal{Y}_i^t , \mathcal{Y}_i^o , and \mathcal{Y}_i^n gathering coordinates of pixels respectively labeled *Ground*, *Target*, *Obstacle*, and *Unknown*. Besides, we assume with (3.24) that any information collected by the camera from $\mathbb{F}(\mathbf{x}_i^c) \cap \mathbb{S}^{\text{RoI}}$ can be reliably processed by the CVS. Consequently, we assume that \mathcal{Y}_i^g , \mathcal{Y}_i^t , \mathcal{Y}_i^o are subsets of a set gathering the coordinates of reliable pixels defined as

$$\mathcal{Y}_i = \left\{ (n_r, n_c) \in \mathcal{N}^l \mid \frac{1}{1 + \underline{w}} \mathbf{D}_i(n_r, n_c) \leq d_{\max} \right\}. \quad (3.36)$$

Unknown pixels are related to elements of the environment that cannot be reliably identified. Thus, the pixels $(n_r, n_c) \in \mathcal{Y}_i^n$ may not be used to estimate a target location since the UAV does not know what has been observed. Nonetheless, we propose to exploit \mathcal{Y}_i^n to approximate the portion of the ground $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$ that has been hidden by unidentified objects. For that purpose, we assume that $\mathcal{Y}_i^n \subset \mathcal{Y}_i$. Consequently, all pixels $(n_r, n_c) \notin \mathcal{Y}_i$ are, in fact, not labeled and not considered in the following chapters.

This section introduces three different assumptions relative to the behavior of the classifier when it processes the information collected by the camera at one time instant. Each of the assumptions will lead to a variant of the estimation algorithm for the location of targets and obstacles detailed in Chapter 4 and Appendix A.1.

With Assumption 4 below, we assume that if a pixel belongs to \mathcal{Y}_i^t , to \mathcal{Y}_i^o , or to \mathcal{Y}_i^g , then all light rays illuminating that pixel come from the same target, from the same obstacle, or from the ground.

Assumption 4. *If a pixel $(n_r, n_c) \in \mathcal{Y}_i^t$ then all light rays illuminating this pixel stem from the same target, i.e.,*

$$\exists j \in \mathcal{N}^t, \forall \mathbf{v} \in \mathcal{V}_i(n_r, n_c), \rho(\mathbf{x}_i^c, \mathbf{v}) = d_{\mathbf{v}}(\mathbf{x}_i^c, \mathbb{S}^t(\mathbf{x}_j^t)). \quad (3.37)$$

Similarly, if $(n_r, n_c) \in \mathcal{Y}_i^o$, then all light rays illuminating this pixel stem from the same obstacle

$$\exists m \in \mathcal{N}^o, \forall \mathbf{v} \in \mathcal{V}_i(n_r, n_c), \rho(\mathbf{x}_i^c, \mathbf{v}) = d_{\mathbf{v}}(\mathbf{x}_i^c, \mathbb{S}_m^o), \quad (3.38)$$

and if $(n_r, n_c) \in \mathcal{Y}_i^g$, then all light rays illuminating this pixel stem from the ground,

$$\forall \mathbf{v} \in \mathcal{V}_i(n_r, n_c), \rho(\mathbf{x}_i^c, \mathbf{v}) = d_{\mathbf{v}}(\mathbf{x}_i^c, \mathbb{X}_g). \quad (3.39)$$

Moreover, in the three cases,

$$\exists \mathbf{v} \in \mathcal{V}_i(n_r, n_c) \text{ such that } \mathbf{D}_i^0(n_r, n_c) = \rho(\mathbf{x}_i^c, \mathbf{v}). \quad (3.40)$$

According to Assumption 4, if a pixel is illuminated by light-rays stemming from different entities, it must belong to \mathcal{Y}_i^n .

With Assumption 5 below, we assume that if a pixel belongs to \mathcal{Y}_i^t , to \mathcal{Y}_i^o , or to \mathcal{Y}_i^g , then at least one light ray illuminating that pixel come from a target, from an obstacle, or from the ground. Moreover, the noise-free distance $\mathbf{D}_i^0(n_r, n_c)$ corresponds to the distance along that light ray.

Assumption 5. *If a pixel $(n_r, n_c) \in \mathcal{Y}_i^t$, then at least one light ray stems from a target, i.e.,*

$$\exists j \in \mathcal{N}^t, \exists \mathbf{v} \in \mathcal{V}_i^t(n_r, n_c), \rho(\mathbf{x}_i^c, \mathbf{v}) = d_{\mathbf{v}}(\mathbf{x}_i^c, \mathbb{S}^t(\mathbf{x}_j^t)), \quad (3.41)$$

with

$$\mathcal{V}_i^t(n_r, n_c) = \{\mathbf{v} \in \mathcal{V}_i(n_r, n_c) \mid \exists j \in \mathcal{N}^t, \rho(\mathbf{x}_i^c, \mathbf{v}) = d_{\mathbf{v}}(\mathbf{x}_i^c, \mathbb{S}^t(\mathbf{x}_j^t))\}, \quad (3.42)$$

and $\mathbf{D}_i^0(n_r, n_c) = \rho(\mathbf{x}_i^c, \mathbf{v})$ along some unknown vector $\mathbf{v} \in \mathcal{V}_i^t(n_r, n_c)$. Similarly, if $(n_r, n_c) \in \mathcal{Y}_i^o$, then

$$\exists m \in \mathcal{N}^o, \exists \mathbf{v} \in \mathcal{V}_i^o(n_r, n_c), \rho(\mathbf{x}_i^c, \mathbf{v}) = d_{\mathbf{v}}(\mathbf{x}_i^c, \mathbb{S}_m^o), \quad (3.43)$$

with

$$\mathcal{V}_i^o(n_r, n_c) = \{\mathbf{v} \in \mathcal{V}_i(n_r, n_c) \mid \exists m \in \mathcal{N}^o, \rho(\mathbf{x}_i^c, \mathbf{v}) = d_{\mathbf{v}}(\mathbf{x}_i^c, \mathbb{S}_m^o)\}, \quad (3.44)$$

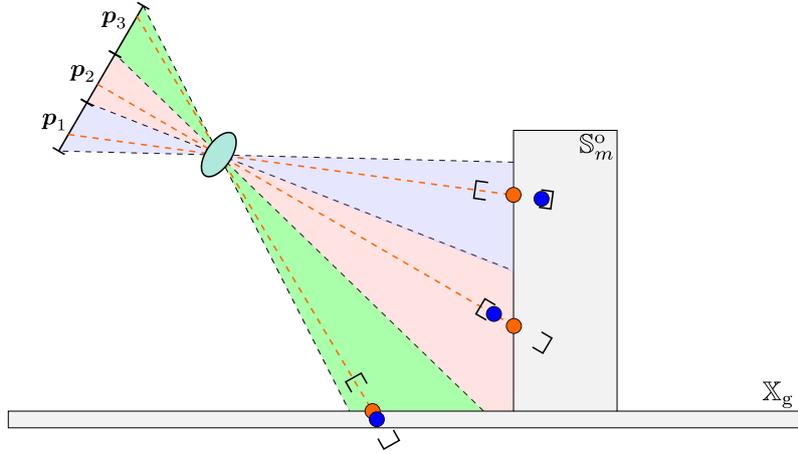


Figure 3.8 – According to Assumption 4, if pixel $p_1 \in \mathcal{Y}_i^o$, then all light rays illuminating p_1 are assumed to stem from an obstacle. If pixel $p_3 \in \mathcal{Y}_i^g$, then all light rays illuminating p_3 are assumed to stem from \mathbb{X}_g . If $p_2 \in \mathcal{Y}_i^n$, then all light rays illuminating p_2 are assumed to stem from a different entities. The orange lines represent the light rays v used to obtain the noise-free distance \mathbf{D}_i^0 (orange dots) for each of the 3 pixels. The blue dots correspond to $\mathbf{D}_i(p_\ell)$ and the black intervals are $[\mathbf{D}_i](p_\ell)$, with $\ell \in \{1, 2, 3\}$, see (3.35).

and $\mathbf{D}_i^0(n_r, n_c) = \rho(\mathbf{x}_i^c, v)$ along some unknown vector $v \in \mathcal{V}_i^o(n_r, n_c)$. Finally, if $(n_r, n_c) \in \mathcal{Y}_i^g$, then

$$\exists v \in \mathcal{V}_i^g(n_r, n_c), \rho(\mathbf{x}_i^c, v) = d_v(\mathbf{x}_i^c, \mathbb{X}_g), \quad (3.45)$$

with

$$\mathcal{V}_i^g(n_r, n_c) = \{v \in \mathcal{V}_i(n_r, n_c) \mid \rho(\mathbf{x}_i^c, v) = d_v(\mathbf{x}_i^c, \mathbb{X}_g)\}, \quad (3.46)$$

and $\mathbf{D}_i^0(n_r, n_c) = \rho(\mathbf{x}_i^c, v)$ along the corresponding $v \in \mathcal{V}_i^g(n_r, n_c)$.

With Assumption 6 below, we assume that if a pixel belongs to \mathcal{Y}_i^t , to \mathcal{Y}_i^o , or to \mathcal{Y}_i^g , then the light ray illuminating the center of its corresponding CCD cell stems from a target, from an obstacle, or from the ground. In other words, with Assumption 6, the portion of the RoI that is observed by a pixel is summarized by the entity (target, obstacle, or ground) that has illuminated its center. Moreover, the noise-free distance $\mathbf{D}_i^0(n_r, n_c)$ corresponds to the distance along that light ray.

Assumption 6. If a pixel $(n_r, n_c) \in \mathcal{Y}_i^t$, then

$$\exists j \in \mathcal{N}^t, \rho(\mathbf{x}_i^c, v^c(n_r, n_c)) = d_{v^c(n_r, n_c)}(\mathbf{x}_i^c, \mathbb{S}^t(\mathbf{x}_j^t)) \quad (3.47)$$

where $v^c(n_r, n_c)$ is the the light ray illuminating the center of pixel (n_r, n_c) . Similarly, if $(n_r, n_c) \in \mathcal{Y}_i^o$, then

$$\exists m \in \mathcal{N}^o, \rho(\mathbf{x}_i^c, v^c(n_r, n_c)) = d_{v^c(n_r, n_c)}(\mathbf{x}_i^c, \mathbb{S}_m^o). \quad (3.48)$$

Finally, if $(n_r, n_c) \in \mathcal{Y}_i^g$, then

$$\rho(\mathbf{x}_i^c, v^c(n_r, n_c)) = d_{v^c(n_r, n_c)}(\mathbf{x}_i^c, \mathbb{X}_g). \quad (3.49)$$

Moreover, one has $\mathbf{D}_i^0(n_r, n_c) = \rho(\mathbf{x}_i^c, v^c(n_r, n_c))$.

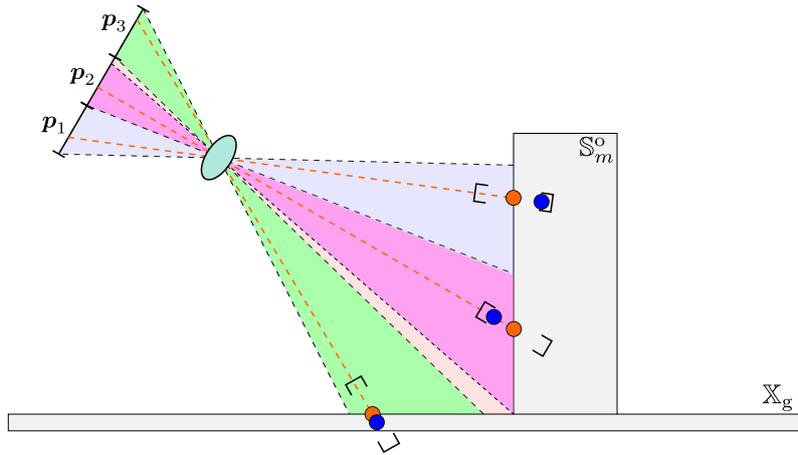


Figure 3.9 – According to Assumption 5, if pixel $p_1 \in \mathcal{Y}_i^o$, then at least one light ray illuminating p_1 stems from an obstacle. If pixel $p_3 \in \mathcal{Y}_i^g$, then at least one light ray illuminating p_3 stems from \mathbb{X}_g . If $p_2 \in \mathcal{Y}_i^n$, nothing can be said about light rays illuminating p_2 . If $p_2 \in \mathcal{Y}_i^o$ or \mathcal{Y}_i^g , then at least one light ray illuminating p_2 is assumed to stem from an obstacle or the ground respectively. Assuming $p_2 \in \mathcal{Y}_i^o$, then the purple set represents $\mathcal{V}_i^o(p_2)$. The orange lines represent the light rays v used to obtain the noise-free distance \mathbf{D}_i^0 (orange dots) for each of the 3 pixels. The blue dots correspond to $\mathbf{D}_i(p_\ell)$ and the black intervals are $[\mathbf{D}_i](p_\ell)$, with $\ell \in \{1, 2, 3\}$, see (3.35).

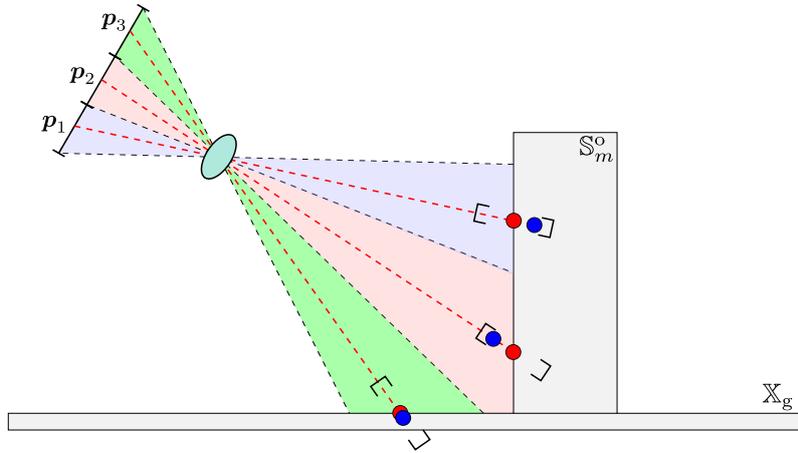


Figure 3.10 – According to Assumption 6, if pixel $p_1 \in \mathcal{Y}_i^o$, then the light ray illuminating the center of p_1 stems from an obstacle. If pixel $p_3 \in \mathcal{Y}_i^g$, then the light ray illuminating the center of p_3 stems from \mathbb{X}_g . If $p_2 \in \mathcal{Y}_i^o$, then the light ray illuminating the center of p_2 stems from an obstacle. The red lines represent the light rays $v^c(p_\ell)$ illuminating the center of p_ℓ that has been used to obtain $\mathbf{D}_i^0(p_\ell)$ (red dots), with $\ell \in \{1, 2, 3\}$. The blue dots correspond to $\mathbf{D}_i(p_\ell)$ and the black intervals are $[\mathbf{D}_i](p_\ell)$, see (3.35).

Each of these assumptions leads to a different way to exploit the information from a labeled pixel. Therefore, during the target localization process, different variants of the set-membership estimators have to be considered depending on the adopted assumption.

Assumptions 4 and 5 appear to be the most realistic, and provide less information related to distance measurements. In Chapter 4, only Assumption 4 is considered. The estimators exploiting Assumption 5 and Assumption 6 are presented in Appendix A.1.

3.5.3 . Target detection and identification

The array of pixel labels \mathbf{L}_i represents a useful source of information to detect the presence of a target within the FoV of UAV i . If \mathcal{Y}_i^t is not empty, then there exists at least one target within $\mathbb{F}(\mathbf{x}_i^u)$, regardless of the assumption chosen to interpret pixels labeled *Target*. Nevertheless, the label carries no information about the identity of the target. Therefore, \mathcal{Y}_i^t can only be used to describe an area of the RoI containing at least one target but whose identity is unknown.

In addition to \mathbf{I}_i , \mathbf{L}_i , and \mathbf{D}_i , the CVS also provides a list $\mathcal{D}_i^t \subset \mathcal{N}^t$ of target indexes relative to the targets that have been identified. The set \mathcal{D}_i^t may be empty when no target is identified in the FoV of UAV i . For each target j identified by UAV i , *i.e.*, $j \in \mathcal{D}_i^t$, an axis-aligned box $[\mathcal{Y}_{i,j}^t]$ locating the target in \mathbf{I}_i is also provided. Such a box can be obtained by combining a target detection algorithm, *e.g.*, (Redmon et al., 2016), with target identification approaches as in (Liu et al., 2022a). To model the behavior of the target detection algorithm in the CVS and exploit the information carried by $[\mathcal{Y}_{i,j}^t]$ for each $j \in \mathcal{D}_i^t$, consider the set $\mathcal{Y}_{i,j}^t \subset \mathcal{Y}_i^t$ containing all pixels of \mathcal{Y}_i^t associated to target j only. Note that $\mathcal{Y}_{i,j}^t$ is **not** provided by the CVS. We introduce $\mathcal{Y}_{i,j}^t$ only to define the following assertions.

According to Assumption 4, $\mathcal{Y}_{i,j}^t$ can be defined as

$$\mathcal{Y}_{i,j}^t = \{(n_r, n_c) \in \mathcal{Y}_i^t \mid \forall \mathbf{v} \in \mathcal{V}_i(n_r, n_c), \rho(\mathbf{x}_i^c, \mathbf{v}) = d_v(\mathbf{x}_i^c, \mathbb{S}^t(\mathbf{x}_j^t))\}. \quad (3.50)$$

The above assertion can be easily adapted for Assumption 5 and 6. We assume that if target j is identified, then $\mathcal{Y}_{i,j}^t$ cannot be empty, *i.e.*,

$$j \in \mathcal{D}_i^t \Rightarrow \mathcal{Y}_{i,j}^t \neq \emptyset. \quad (3.51)$$

Moreover, the CVS is assumed to be tuned in such a way that $[\mathcal{Y}_{i,j}^t]$ either contains $\mathcal{Y}_{i,j}^t$, *i.e.*,

$$j \in \mathcal{D}_i^t \Rightarrow \mathcal{Y}_{i,j}^t \subset [\mathcal{Y}_{i,j}^t], \quad (3.52)$$

or intersect $\mathcal{Y}_{i,j}^t$, *i.e.*,

$$j \in \mathcal{D}_i^t \Rightarrow \mathcal{Y}_{i,j}^t \cap [\mathcal{Y}_{i,j}^t] \neq \emptyset. \quad (3.53)$$

Assumption (3.53) is relatively mild and may well reflect the behavior of detection algorithms, which are usually not able to ensure (3.52). Consequently, in what follows, only (3.53) will be considered.

3.6 . Assumption on observed targets

This section introduces an observation condition that, if satisfied, allows the exploitation of *Ground*-labeled pixels as negative measurements. To simplify notations, the time index is again omitted in this section.

We recall that the location \mathbf{x}_j^{tg} for a target j does not necessarily belong to its shape (see Figure 3.11). Thus, for a pixel (n_r, n_c) such that $\mathbf{p}_c(\mathbf{x}_i^u, \mathbf{x}_j^{\text{tg}}) \in (n_r, n_c)$, none of the three assumptions presented in Section 3.5.2 is useful to predict if (n_r, n_c) would be labeled *Target*, if the target is not hidden, *Obstacle*, if it is behind an obstacle, *Unknown*, or *Ground* since \mathbf{x}_j^{tg} belong to \mathbb{X}_g . Consequently, another assumption is needed.

Assumption 7. *If a target j is such that $\mathbf{x}_j^{\text{tg}} \in \mathbb{F}(\mathbf{x}_i^u)$, then the half-open segment $[\mathbf{x}_i^c, \mathbf{x}_j^{\text{tg}}[$ intersects $\mathbb{S}^t(\mathbf{x}_j^t)$, i.e.,*

$$\mathbf{x}_j^{\text{tg}} \in \mathbb{F}(\mathbf{x}_i^u) \implies [\mathbf{x}_i^c, \mathbf{x}_j^{\text{tg}}[\cap \mathbb{S}^t(\mathbf{x}_j^t) \neq \emptyset. \quad (3.54)$$

The validity of (3.54) depends on the camera orientation and the characteristics of the target shape $\mathbb{S}^t(\mathbf{x}_j^t)$. This assumption is reasonable for many targets such as cars or trucks, provided that they are observed from a location sufficiently above the target, with a camera oriented towards the ground.

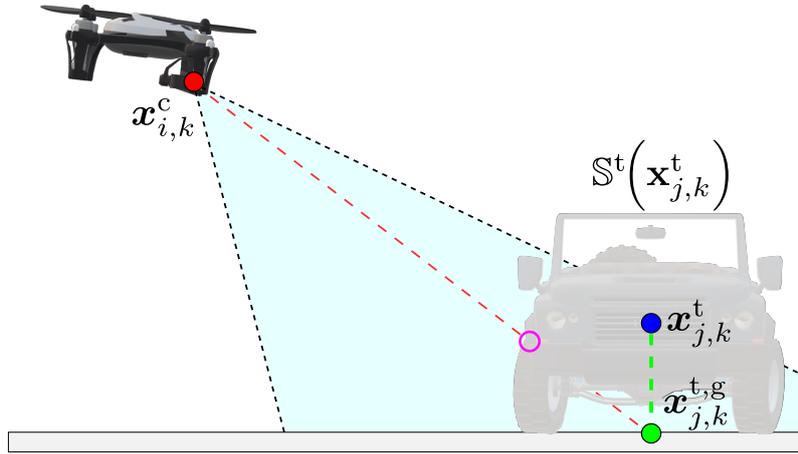


Figure 3.11 – Illustration of (3.54). The blue dot represents \mathbf{x}_j^t and the green dot represents \mathbf{x}_j^{tg} . The dashed-red line is the line-of-sight $[\mathbf{x}_i^c, \mathbf{x}_j^{\text{tg}}[$. The purple dot represents the part of $\mathbb{S}^t(\mathbf{x}_j^t)$ seen by the UAV.

Under Assumption 7 and Assumption 4 applied to *Ground*-labeled pixels, if $\mathbf{x}_j^{\text{tg}} \in \mathbb{F}(\mathbf{x}_i^u)$, then the pixel (n_r, n_c) such that $\mathbf{p}_c(\mathbf{x}_i^u, \mathbf{x}_j^{\text{tg}}) \in (n_r, n_c)$ cannot be labeled *Ground* since there exists at least one light ray that does not stem from the ground. In addition, one can remark that such a property cannot be guaranteed with Assumption 5 or 6. Therefore, Assumption 4 must be considered for *Ground*-labeled pixels to make use of (3.54).

3.7 . Problem formulation

Assume that the information available to UAV i before time t_k is gathered in the set $\mathcal{I}_{i,k-1}$. $\mathcal{I}_{i,k-1}$ contains, among others, a list $\mathcal{L}_{i,k-1}^t$ of indexes of targets already identified. For each target $j \in \mathcal{L}_{i,k-1}^t$,

we assume that UAV i has access to a set estimate $\mathbb{X}_{i,j,k-1}^t \subset \mathbb{X}_g$ of all possible target locations $\mathbf{x}_{j,k-1}^{tg}$ which are consistent with $\mathcal{I}_{i,k-1}$. It has also access to a set $\bar{\mathbb{X}}_{i,k-1}^t$ containing all locations of targets still to be identified.

At time t_k , the camera and CVS of UAV i provides an image $\mathbf{I}_{i,k}$, a depth-map $\mathbf{D}_{i,k}$, and an array of labels $\mathbf{L}_{i,k}$. In addition, the CVS may also provide a list $\mathcal{D}_{i,k}^t$ of identified targets and a list $\mathcal{B}_{i,k}^t = \left\{ \left[\mathcal{Y}_{i,j,k}^t \right] \right\}_{j \in \mathcal{D}_{i,k}}$ of associated bounding boxes. The information available at time t_k is then $\mathcal{I}_{i,k|k} = \mathcal{I}_{i,k-1} \cup \left\{ \mathbf{I}_{i,k}, \mathbf{D}_{i,k}, \mathbf{L}_{i,k}, \mathcal{D}_{i,k}^t, \mathcal{B}_{i,k}^t \right\}$.

Using this information UAV i can evaluate an updated list of identified targets $\mathcal{L}_{i,k|k}^t = \mathcal{L}_{i,k-1}^t \cup \mathcal{D}_{i,k}^t$. Then, for each targets $j \in \mathcal{L}_{i,k|k}^t$ it has to characterize the set $\mathbb{X}_{i,j,k|k}^t \subset \mathbb{X}_g$ of all possible target locations $\mathbf{x}_{j,k}^{tg}$ which are consistent with $\mathcal{I}_{i,k|k}$. UAV i has also to update the estimate $\bar{\mathbb{X}}_{i,k-1}^t$ to get $\bar{\mathbb{X}}_{i,k|k}^t$. Then, UAV i broadcasts some updated information such as $\mathcal{L}_{i,k|k}^t$, $\left\{ \mathbb{X}_{i,j,k|k}^t \right\}_{j \in \mathcal{L}_{i,k|k}^t}$ and $\bar{\mathbb{X}}_{i,k|k}^t$ to its neighbors with indexes in $\mathcal{N}_{i,k}$ and receives similar information from them. With this new information, UAV i can further update its list of identified targets to get $\mathcal{L}_{i,k}^t$, the set estimates $\mathbb{X}_{i,j,k}^t$ for each $j \in \mathcal{L}_{i,k}^t$, which are gathered in $\mathcal{X}_{i,k}^t = \left\{ \mathbb{X}_{i,j,k}^t \right\}_{j \in \mathcal{L}_{i,k}^t}$, as well as an updated version $\bar{\mathbb{X}}_{i,k}^t$ of $\bar{\mathbb{X}}_{i,k|k}^t$.

The estimation uncertainty about the localization of both identified and undetected targets may be defined as

$$\Phi \left(\mathcal{X}_{i,k}^t, \bar{\mathbb{X}}_{i,k}^t \right) = \phi \left(\bar{\mathbb{X}}_{i,k}^t \cup \bigcup_{j \in \mathcal{L}_{i,k}^t} \mathbb{X}_{i,j,k}^t \right), \quad (3.55)$$

where $\phi(\mathbb{X})$ is the area of the set $\mathbb{X} \subset \mathbb{R}^2$.

Our aim is to obtain the smallest set estimates from the available measurements and to design a trajectory for each UAV in order to minimize (3.55).

3.8 . Conclusion

This chapter formalizes the CSAT problem with ground targets evolving within an unknown but structured environment. A fleet of cooperative UAVs is deployed to search, identify, and localize those targets. For that purpose, each UAV embeds a CVS to collect and process information from the environment. Similarly to (Goldhoorn et al., 2018; Meera et al., 2019; Kenmogne Fokam, 2019), and unlike (Kuhlman et al., 2017; Zhu et al., 2021; Zhao et al., 2022; Ibenthal et al., 2023; Zhang et al., 2024), no idealized CVS is considered to solve our CSAT problem. Instead, each UAV must exploit available CVS measurements consisting of a labeled image, a depth map, and a list of bounding boxes associated to identified targets, to estimate the location of the targets. The target location estimator that exploits these CVS measurements is presented in Chapter 4. Additionally, the environment is *a priori* unknown. Thus, each UAV must exploit the CVS measurements to gain knowledge regarding the shape and location of the obstacles. This environmental knowledge is then used in Chapter 5 with an approach similar to the one presented in (Meera et al., 2019) to better design the updated trajectory of each UAV by predicting the portion of the RoI that will be masked by the detected obstacles.

How these CVS measurements are obtained is often hard to model since, nowadays, they often involve deep learning algorithms, as in (Cao et al., 2017; Luo et al., 2022). Therefore, in most cases, there are no simple measurement models to represent the computer vision algorithms. To address this issue, a camera model is introduced in Section 3.4, and assumptions modeling how the CVS measurements can be related to the RoI are introduced in Section 3.5. The purpose of those assumptions is to make the available CVS measurements exploitable by the target location estimator presented in Chapter 4. We use the pinhole model to model the camera. The depth map model (3.31) is consistent with this camera model. Then, three different assumptions are introduced to relate a labeled pixel to the RoI. Each of those assumptions has its advantages and drawbacks. If the UAVs are not too far from the RoI, then assumptions 4 and 6 seem to be the simplest assumptions to consider. Nonetheless, Assumption 4 assumes that as soon as a pixel is labeled *Target*, then that pixel must have been illuminated by only one target. From an aerial point of view, if the UAVs fly high, the targets are small. Consequently, this assumption is a bit unrealistic. Regarding Assumption 6, only the light ray illuminating the center of a labeled pixel is considered. This simplification may become unrealistic for the same reason because a detected target may not be located at the exact center of a *Target*-labeled pixel. Assumption 5 appears to be the most realistic. The subsets \mathcal{V}_i^ℓ , with $\ell \in \{t, g, o\}$, model the fact that if a pixel is labeled, e.g., *Target*, then some light rays must have stemmed from a target. The subset \mathcal{V}_i^t , however, is unknown. Consequently, and as explained in Appendix A.1, there is no difference between Assumptions 4 and 5 regarding the design of the target location estimator. That is why only Assumption 4 is considered in the following chapters.

Assumption (3.54) is instrumental in transforming some of the available CVS measurements into negative information. Besides, this assumption can be easily satisfied. Moreover, Section 3.5.3 reflects the behavior of detection algorithms that, with the target identification algorithm of (Liu et al., 2022a), can provide a reliable target identification. Nevertheless, possible bad evaluations of the bounding box $\left[\mathcal{Y}_{i,j,k}^t \right]$ are possible, as it may be the case when the target detector encounters a situation which has not been well represented in its training data set. In such situations, the Intersection over Union between the bounding box and the ground truth may be very low. Nonetheless, if Assumption (3.53) is satisfied, i.e., there is at least one pixel associated to the identified target which is located within the bounded box, then, a robust target location estimation is still possible with the estimator presented in Chapter 4. Nonetheless, target misidentification, and target non-detection are problems that have not been considered. The target misidentification problem can be handled by the approach developed in (Ibenthal et al., 2020), and False Positives, i.e., a *Target*-labeled pixel that is not illuminated by any target, could be handled by our approach. Nevertheless, target non-detection or False Negatives, i.e., a pixel is labeled *Ground* even though it has been illuminated by a target, is still a problem that is not formalized in this chapter, and not addressed in this thesis.

Chapter 4 - Target localization from CVS measurements

In this chapter, a method to estimate the location of targets from measurements provided by a Computer Vision System (CVS) embedded in a UAV is presented. Using a set-membership approach, the aim of the proposed estimator is to characterize sets that are guaranteed to contain the location of a target, and sets that are guaranteed to contain no target location. For that purpose, the assumptions introduced in Chapter 3 will be used.

This chapter considers the part of the Cooperative Search, Acquisition, and Tracking (CSAT) problem devoted to the estimation process and the update of the knowledge of a UAV about the location targets using the CVS measurements. The design of a trajectory to track those targets is addressed in Chapter 5.

The notations used in this chapter are first recalled in Section 4.1 before a formulation of the estimation problem studied in this chapter. Then, Section 4.2 presents a method to exploit the measurements provided by the CVS to characterize various set estimates related to the location of identified targets. Section 4.3 presents a method to characterize set estimates containing no target location. Section 4.4 proposes an approach to estimate the portion of the ground that has been occluded by an obstacle. Those sets are then used to update the knowledge each UAV has about the possible locations of the targets by using a prediction-correction approach that is presented in Section 4.5. Section 4.6 describes the way CVS measurements are used to build the map of the Region of Interest (RoI). Then, some simulations and results are presented in Section 4.7 before concluding.

Appendix A.2 describes how one may compute the introduced set estimates.

To simplify the notation, the time index is omitted in the Sections 4.2, 4.3, and 4.4.

4.1 . Problem formulation

A fleet of N^u UAVs is dispatched within an RoI \mathbb{X}_0 which contains N^t mobile ground targets and is cluttered with N^o static obstacles of unknown shape and location.

Just before time t_k , assume that UAV $i \in \mathcal{N}^u$ has access to some knowledge about some previously identified targets and the environment. This knowledge consists of a list $\mathcal{L}_{i,k-1}^t$ of indexes of targets already identified. For each $j \in \mathcal{L}_{i,k-1}^t$, UAV i has access to a set estimate $\mathbb{X}_{i,j,k-1}^t \subset \mathbb{X}_g$ containing the location $\mathbf{x}_{j,k-1}^t$ of target j at time t_{k-1} . The list of sets $\mathcal{X}_{i,k-1}^t = \left\{ \mathbb{X}_{i,j,k-1}^t \right\}_{j \in \mathcal{L}_{i,k-1}^t}$ gathers all the knowledge UAV i has about the identified targets. It has also access to a set $\overline{\mathbb{X}}_{i,k-1}^t$ containing all locations where potential unidentified targets may still be present. Lastly, UAV i has access to a set $\mathbb{X}_{i,k-1}^o$ representing areas around previously observed obstacles where no target can be. Those previously observed obstacles may also have been mapped through the Occupancy Elevation Map (OEM) $\mathcal{M}_{i,k-1}$.

At time t_k , UAV i , with state $\mathbf{x}_{i,k}^u$, observes a subset $\mathbb{F}(\mathbf{x}_{i,k}^u)$ of its environment using a camera to get an image $\mathbf{I}_{i,k}$. Then, the embedded CVS processes $\mathbf{I}_{i,k}$ to provide a depth map $\mathbf{D}_{i,k}$, an array of pixel labels $\mathbf{L}_{i,k}$, a list $\mathcal{D}_{i,k}^t$ of targets identified at time t_k , and a list $\mathcal{B}_{i,k}^t = \left\{ \left[\mathcal{Y}_{i,j,k}^t \right] \right\}_{j \in \mathcal{D}_{i,k}^t}$ of bounding

boxes associated to those identified targets.

The problem addressed in this chapter is the exploitation of the measurements provided by the CVS using the assumptions presented in Chapter 3 to localize the targets. For that purpose, if $j \in \mathcal{D}_{i,k}^t$, a set estimate $\mathbb{X}_{i,j,k}^{t,m}$ containing the location $\mathbf{x}_{j,k}^t$ of a target j is evaluated. A set estimate $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$ is also characterized using the pixels labeled *Ground* such that $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$ cannot contain any target location. Similarly, pixels labeled *Obstacles* are used in two ways. The first is to characterize a set $\mathbb{X}_{i,k}^o$ that can never contain any target location due to the proximity of an obstacle. The second is to update the OEM.

Then, once characterized, these set estimates as well as the information broadcast by potential neighbors are used to update the list $\mathcal{L}_{i,k-1}^t$ and the sets $\mathcal{X}_{i,k-1}^t$, $\bar{\mathbb{X}}_{i,k-1}^t$ and $\mathbb{X}_{i,k-1}^o$ to get $\mathcal{L}_{i,k}^t$, $\mathcal{X}_{i,k}^t$, $\bar{\mathbb{X}}_{i,k}^t$ and $\mathbb{X}_{i,k}^o$.

4.2 . Exploiting CVS information to localize an identified target

Consider that a target $j \in \mathcal{D}_i^t$ is identified by UAV i at time t_k . The CVS provides a depth map \mathbf{D}_i , a list \mathcal{Y}_i^t of pixels labeled *Target*, and a bounding box $[\mathcal{Y}_{i,j}^t]$. According to (3.51), since the target has been identified, then at least one pixel contained in \mathcal{Y}_i^t is associated to target j . Therefore, by combining $[\mathcal{Y}_{i,j}^t]$ with \mathcal{Y}_i^t and \mathbf{D}_i , UAV i is able to characterize a set estimate $\mathbb{X}_{i,j}^{t,m}$ containing the target location $\mathbf{x}_j^{t,g}$.

Chapter 3 has introduced three assumptions to describe the way measurements provided by the CVS can be related to the environment. This chapter only considers Assumption 4. Appendix A.1 presents variants of that estimator that one would obtain by considering the other assumptions.

We start with a preliminary result : Assume that a point \mathbf{x} belonging to the shape $\mathbb{S}^t(\mathbf{x}_j^t)$ of a target j is available, then, using the following proposition, it is possible to characterize a set containing the location $\mathbf{x}_j^{t,g}$ of target j .

Proposition 8. *For all $j \in \mathcal{N}^t$, if $\mathbf{x} \in \mathbb{S}^t(\mathbf{x}_{j,k}^t)$, then $\mathbf{x}_{j,k}^{t,g} \in \mathbb{N}_g(\{\mathbf{p}_g(\mathbf{x})\}, r^t)$.*

Démonstration. Consider some $\mathbf{x} \in \mathbb{S}^t(\mathbf{x}_j^t)$. According to (3.4), we have $\mathbb{S}^t(\mathbf{x}_j^t) \subset \mathbb{C}^t(\mathbf{x}_j^{t,g})$, where $\mathbb{C}^t(\mathbf{x}_j^{t,g})$ is a vertical circular right cylinder of height h^t and radius r^t with basis centered in $\mathbf{x}_j^{t,g}$. Thus, we also have $\mathbf{p}_g(\mathbb{S}^t(\mathbf{x}_j^t)) \subset \mathbf{p}_g(\mathbb{C}^t(\mathbf{x}_j^{t,g}))$. We recall that $\mathbf{p}_g(\mathbb{C}^t(\mathbf{x}_j^{t,g}))$ is a disc of radius r^t and center $\mathbf{x}_j^{t,g}$. Another notation of this disc is $\mathbb{N}_g(\{\mathbf{x}_j^{t,g}\}, r^t)$. Therefore, we have $\mathbf{p}_g(\mathbf{x}) \in \mathbb{N}_g(\{\mathbf{x}_j^{t,g}\}, r^t)$.

Consequently, one has $\|\mathbf{p}_g(\mathbf{x}) - \mathbf{x}_j^{t,g}\| \leq r^t$. This is equivalent to have $\|\mathbf{x}_j^{t,g} - \mathbf{p}_g(\mathbf{x})\| \leq r^t$. Consequently, we deduce that $\mathbb{N}_g(\{\mathbf{p}_g(\mathbf{x})\}, r^t)$ contains $\mathbf{x}_j^{t,g}$. \square

Consequently, if a target j is identified, it suffices to determine $\mathbf{x} \in \mathbb{S}^t(\mathbf{x}_j^t)$ to be able to build a set estimate $\mathbb{N}_g(\{\mathbf{p}_g(\mathbf{x})\}, r^t)$ containing $\mathbf{x}_j^{t,g}$. This will be done in what follows.

If a pixel $(n_r, n_c) \in \mathcal{Y}_i^t$, i.e., is labeled *Target*, then, according to Assumption 4, all light rays contributing to its illumination are stemming from a single target $\ell \in \mathcal{N}^t$. Assumption 4 does not provide any information about ℓ . Using the depth map measurement $\mathbf{D}_i(n_r, n_c)$ associated to that pixel, our aim is to characterize a set $\mathbb{P}_i((n_r, n_c))$ intersecting the shape of the observed target ℓ .

For any pixel $(n_r, n_c) \in \mathbf{I}_i$, the set

$$\mathbb{P}_i((n_r, n_c)) = \{ \mathbf{x} \in \mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_0 \mid \exists \mathbf{v} \in \mathcal{V}_i(n_r, n_c), d_v(\mathbf{x}_i^c, \{\mathbf{x}\}) \in [\mathbf{D}_i](n_r, n_c) \} \quad (4.1)$$

contains all points in $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_0$ that may have contributed to the illumination of (n_r, n_c) while being at a distance from UAV i consistent with $\mathbf{D}_i(n_r, n_c)$. According to Proposition 9, if $(n_r, n_c) \in \mathcal{Y}_i^t$, then $\mathbb{P}_i((n_r, n_c))$ intersects the shape $\mathbb{S}^t(\mathbf{x}_\ell^t)$ of some target ℓ .

Proposition 9. *If $(n_r, n_c) \in \mathcal{Y}_i^t$, then there exists $\ell \in \mathcal{N}^t$ such that $\mathbb{P}_i((n_r, n_c)) \cap \mathbb{S}^t(\mathbf{x}_\ell^t) \neq \emptyset$.*

Démonstration. If $(n_r, n_c) \in \mathcal{Y}_i^t$, then according to Assumption 4, there exists a target $\ell \in \mathcal{N}^t$ such that for all $\mathbf{v} \in \mathcal{V}_i(n_r, n_c)$, $\rho(\mathbf{x}_i^c, \mathbf{v}) = d_v(\mathbf{x}_i^c, \mathbb{S}^t(\mathbf{x}_\ell^t))$. Therefore, there exist $\mathbf{v} \in \mathcal{V}_i(n_r, n_c)$ and $\mathbf{x}^* \in \mathbb{S}^t(\mathbf{x}_\ell^t) \subset \mathbb{X}_0$ such that $d_v(\mathbf{x}_i^c, \{\mathbf{x}^*\}) = \mathbf{D}_i^0(n_r, n_c)$. According to (3.35), one has $d_v(\mathbf{x}_i^c, \{\mathbf{x}^*\}) \in [\mathbf{D}_i](n_r, n_c)$. In addition, as $(n_r, n_c) \in \mathcal{Y}_i^t \subset \mathcal{Y}_i$, see (3.36), then $\frac{1}{1+w} \mathbf{D}_{i,k}(n_r, n_c) \leq d_{\max}$ and $d_v(\mathbf{x}_i^c, \{\mathbf{x}^*\}) \leq d_{\max}$. Therefore $\mathbf{x}^* \in \mathbb{F}(\mathbf{x}_i^u)$. Finally, from (4.1), one has $\mathbf{x}^* \in \mathbb{P}_i((n_r, n_c))$, and thus $\mathbb{P}_i((n_r, n_c)) \cap \mathbb{S}^t(\mathbf{x}_\ell^t) \neq \emptyset$. \square

Figure 4.1 illustrates Proposition 9 with three pixels labeled *Target*. The orange sets are the $\mathbb{P}_i((n_r, n_c))$ obtained for each pixel, and they contain a point \mathbf{x}^* (in red) belonging to a target (here, the same target) for which the noise-free depth measurement $\mathbf{D}_i^0(n_r, n_c)$ is assumed to have been obtained.

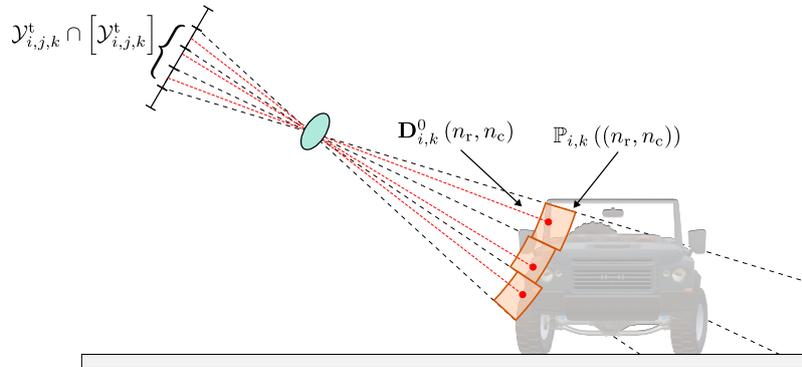


Figure 4.1 – Illustration of the sets $\mathbb{P}_i((n_r, n_c))$ (in orange) for different $(n_r, n_c) \in \mathcal{Y}_{i,j}^t \cap [\mathcal{Y}_{i,j}^t]$ and for a target detected and identified by UAV i

According to (3.51), since target j has been identified by UAV i at time t_{k_i} , the set $\mathcal{Y}_{i,j}^t$ of pixels labeled *Target* associated to target j is not empty. Moreover, from (3.53), one has $\mathcal{Y}_{i,j}^t \cap [\mathcal{Y}_{i,j}^t] \neq \emptyset$. Even if only the set of pixels \mathcal{Y}_i^t such that $\mathcal{Y}_{i,j}^t \subset \mathcal{Y}_i^t$ is available, see Section 3.5.3, using the following Proposition 10, there exists at least one pixel $(n_r, n_c) \in [\mathcal{Y}_{i,j}^t] \cap \mathcal{Y}_i^t$ for which the set $\mathbb{P}_i((n_r, n_c))$ intersects $\mathbb{S}^t(\mathbf{x}_j^t)$.

Proposition 10. *If $j \in \mathcal{D}_i^t$, then $\exists (n_r, n_c) \in [\mathcal{Y}_{i,j}^t] \cap \mathcal{Y}_i^t$ such that $\mathbb{P}_i((n_r, n_c)) \cap \mathbb{S}^t(\mathbf{x}_j^t) \neq \emptyset$.*

Démonstration. If $j \in \mathcal{D}_i^t$, then $\mathcal{Y}_{i,j}^t \neq \emptyset$ and $\mathcal{Y}_{i,j}^t \cap [\mathcal{Y}_{i,j}^t] \neq \emptyset$. Combining (3.51), (3.53), there exists a pixel $(n_r, n_c) \in [\mathcal{Y}_{i,j}^t] \cap \mathcal{Y}_i^t$ such that $(n_r, n_c) \in \mathcal{Y}_{i,j}^t$. Then, according to (3.50) and Proposition 9, we deduce $\mathbb{P}_i((n_r, n_c)) \cap \mathbb{S}^t(\mathbf{x}_j^t) \neq \emptyset$. \square

The pixel (n_r, n_c) such that $\mathbb{P}_i((n_r, n_c)) \cap \mathbb{S}^t(\mathbf{x}_j^t) \neq \emptyset$ exists, but it is only known to belong to $\mathcal{Y}_i^t \cap [\mathcal{Y}_{i,j}^t]$. To get a set intersecting $\mathbb{S}^t(\mathbf{x}_j^t)$, one has to consider the union of all sets $\mathbb{P}_i((n_r, n_c))$ over the pixels $(n_r, n_c) \in [\mathcal{Y}_{i,j}^t] \cap \mathcal{Y}_i^t$ to obtain a set $\mathbb{P}_{i,j}^t$ intersecting $\mathbb{S}^t(\mathbf{x}_j^t)$, as reported in Corollary 11 (a direct consequence of Proposition 10).

Corollary 11. *If $j \in \mathcal{D}_i^t$, then the set*

$$\mathbb{P}_{i,j}^t = \bigcup_{(n_r, n_c) \in [\mathcal{Y}_{i,j}^t] \cap \mathcal{Y}_i^t} \mathbb{P}_i((n_r, n_c)) \quad (4.2)$$

is such that $\mathbb{P}_{i,j}^t \cap \mathbb{S}^t(\mathbf{x}_j^t) \neq \emptyset$.

Then, according to Proposition 12, if $\mathbb{P}_{i,j}^t \cap \mathbb{S}^t(\mathbf{x}_j^t) \neq \emptyset$, then $\mathbf{p}_g(\mathbb{P}_{i,j}^t) \cap \mathbf{p}_g(\mathbb{S}^t(\mathbf{x}_j^t)) \neq \emptyset$.

Proposition 12. *Consider the sets \mathbb{A} and \mathbb{B} . If $\mathbb{A} \cap \mathbb{B} \neq \emptyset$, then we have $\mathbf{p}_g(\mathbb{A} \cap \mathbb{B}) \subset \mathbf{p}_g(\mathbb{A}) \cap \mathbf{p}_g(\mathbb{B})$*

Démonstration. Consider $\mathbf{x} \in \mathbb{A} \cap \mathbb{B}$. Then, we have $\mathbf{x} \in \mathbb{A}$, $\mathbf{x} \in \mathbb{B}$ and $\mathbf{p}_g(\mathbf{x}) \in \mathbf{p}_g(\mathbb{A} \cap \mathbb{B})$. As $\mathbf{x} \in \mathbb{A}$, we have $\mathbf{p}_g(\mathbf{x}) \in \mathbf{p}_g(\mathbb{A})$. We have the same for \mathbb{B} . Consequently, $\mathbf{p}_g(\mathbb{A} \cap \mathbb{B}) \subset \mathbf{p}_g(\mathbb{A}) \cap \mathbf{p}_g(\mathbb{B})$. \square

$\mathbb{P}_{i,j}^t$ is only known to intersect $\mathbb{S}^t(\mathbf{x}_j^t)$. If $\mathbf{x} \in \mathbb{P}_{i,j}^t \cap \mathbb{S}^t(\mathbf{x}_j^t)$ would be available, one would be able to exploit Proposition 8 to characterize a set estimate $\mathbb{N}_g(\{\mathbf{p}_g(\mathbf{x})\}, r^t)$ for \mathbf{x}_j^{tg} . As \mathbf{x} is only known to belong to $\mathbb{P}_{i,j}^t$, using Proposition 13, one obtains a set estimate $\mathbb{X}_{i,j}^{t,m}$ containing \mathbf{x}_j^{tg} .

Proposition 13. *If $j \in \mathcal{D}_i^t$, then the set estimate*

$$\mathbb{X}_{i,j}^{t,m} = \mathbb{X}_g \cap \bigcup_{\mathbf{x} \in \mathbf{p}_g(\mathbb{P}_{i,j}^t)} \mathbb{N}_g(\{\mathbf{x}\}, r^t) \quad (4.3)$$

is such that $\mathbf{x}_j^{tg} \in \mathbb{X}_{i,j}^{t,m}$.

Démonstration. As $j \in \mathcal{D}_i^t$, we have $\mathbb{P}_{i,j}^t \cap \mathbb{S}_j^t(\mathbf{x}_j^t) \neq \emptyset$. Then, using Proposition 12, we have $\mathbf{p}_g(\mathbb{P}_{i,j}^t) \cap \mathbf{p}_g(\mathbb{S}^t(\mathbf{x}_j^t)) \neq \emptyset$. So, there exists $\mathbf{x} \in \mathbf{p}_g(\mathbb{P}_{i,j}^t) \cap \mathbf{p}_g(\mathbb{S}^t(\mathbf{x}_j^t))$. Consequently, from Proposition 8, one has $\mathbf{x}_j^{tg} \in \mathbb{N}_g(\{\mathbf{x}\}, r^t)$. However, as $\mathbb{P}_{i,j}^t$ is only known to have a non-empty intersection with $\mathbb{S}^t(\mathbf{x}_j^t)$, to build a set estimate for \mathbf{x}_j^{tg} , one has to consider all $\mathbf{x} \in \mathbf{p}_g(\mathbb{P}_{i,j}^t)$ and build the union of $\mathbb{N}_g(\{\mathbf{x}\}, r^t)$ over all $\mathbf{x} \in \mathbf{p}_g(\mathbb{P}_{i,j}^t)$. Then, since \mathbf{x}_j^{tg} is always within \mathbb{X}_g , we deduce that the set $\mathbb{X}_{i,j}^{t,m} = \mathbb{X}_g \cap \bigcup_{\mathbf{x} \in \mathbf{p}_g(\mathbb{P}_{i,j}^t)} \mathbb{N}_g(\{\mathbf{x}\}, r^t)$ contains \mathbf{x}_j^{tg} . \square

In practice, the estimate $\mathbb{X}_{i,j}^{t,m}$ is evaluated as

$$\mathbb{X}_{i,j}^{t,m} = \mathbb{X}_g \cap (\mathbf{p}_g(\mathbb{P}_{i,j}^t) \oplus \mathbb{N}_g(\{\mathbf{0}\}, r^t)), \quad (4.4)$$

which, compared to (4.3), only involves the Minkovski sum of the disc $\mathbb{N}_g(\{\mathbf{0}\}, r^t)$ and the projection of $\mathbb{P}_{i,j}^t$ on \mathbb{X}_g , see Figure 4.2.

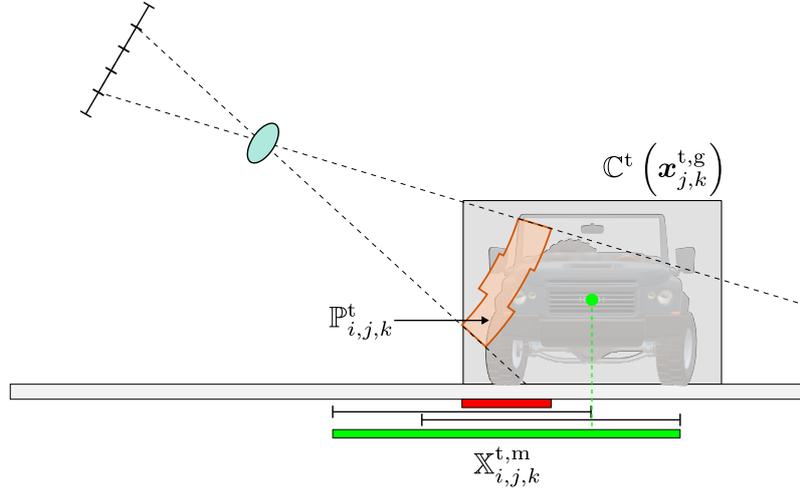


Figure 4.2 – Estimation of the location of target j : The set $\mathbb{P}_{i,j}^t$ is projected on the ground (red set). Then $\mathbf{p}_g(\mathbb{C}^t)$ (horizontal circles represented by the black lines) is used to build $\mathbb{X}_{i,j}^{t,m}$ (green set). The green dot represents \mathbf{x}_j^t .

4.3 . Exploiting CVS information to characterize a target-free area

An approach to characterize a subset of \mathbb{X}_g that contains no target location is presented in this section. This is achieved by processing the pixels labeled *Ground*, in Section 4.3.1, *Obstacle*, in Section 4.3.2, or *Target*, in Section 4.3.3.

4.3.1 . Using pixels labeled Ground

This section presents an approach to characterize a subset of $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$, *i.e.*, the intersection of the FoV with the ground, containing no location of any target $j \in \mathcal{N}^t$.

For that purpose, we first introduce the set

$$\mathbb{P}_i^g((n_r, n_c)) = \{\mathbf{x} \in \mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g \mid \mathbf{p}_c(\mathbf{x}_i^u, \mathbf{x}) \in (n_r, n_c)\} \quad (4.5)$$

that contains all points in $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$ whose projection on the CCD array $\mathbf{p}_c(\mathbf{x}_i^u, \mathbf{x})$ belongs to (n_r, n_c) , see (3.29). $\mathbb{P}_i^g((n_r, n_c))$, illustrated in Figure 3.7, represents the portion of \mathbb{X}_g that would be observed by pixel (n_r, n_c) in absence of obstacles and targets.

According to Assumption 4, pixels in \mathcal{Y}_i^g , *i.e.*, labeled *Ground*, are such that only points in \mathbb{X}_g have contributed to their illumination. Consequently, the set

$$\mathbb{P}_i^g(\mathcal{Y}_i^g) = \{\mathbf{x} \in \mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g \mid \mathbf{p}_c(\mathbf{x}_i^u, \mathbf{x}) \in \mathcal{Y}_i^g\} \quad (4.6)$$

of points of \mathbb{X}_g whose projection onto the CCD array belongs to a pixel labeled *Ground* cannot contain the location of a target, see Proposition 14.

Proposition 14. For all $j \in \mathcal{N}^t$, $\mathbf{x}_j^{t,g} \notin \mathbb{P}_i^g(\mathcal{Y}_i^g)$.

Démonstration. First, if $\mathbf{x}_j^{t,g} \notin \mathbb{F}(\mathbf{x}_i^u)$, then $\mathbf{x}_j^{t,g} \notin \mathbb{P}_i^g(\mathcal{Y}_i^g)$, by definition of $\mathbb{P}_i^g(\mathcal{Y}_i^g)$. Consider now $\mathbf{x}_j^{t,g} \in \mathbb{F}(\mathbf{x}_i^u)$. According to (3.29), there exists $(n_r, n_c) \in \mathcal{N}^l$ such that $\mathbf{p}_c(\mathbf{x}_i^u, \mathbf{x}_j^{t,g}) \in (n_r, n_c)$. Moreover, according to (3.54), there exists an half-open segment $[\mathbf{x}_i^c, \mathbf{x}_j^{t,g}[$ such that $[\mathbf{x}_i^c, \mathbf{x}_j^{t,g}[\cap \mathbb{S}^t(\mathbf{x}_j^t) \neq \emptyset$ while having $[\mathbf{x}_i^c, \mathbf{x}_j^{t,g}[\cap \mathbb{X}_g = \emptyset$. Consider the unit vector $\mathbf{v} \in \mathcal{V}_i(n_r, n_c)$ such that \mathbf{v} is colinear with $[\mathbf{x}_i^c, \mathbf{x}_j^{t,g}[$. Therefore, $\rho(\mathbf{x}_i^c, \mathbf{v}) \neq d_v(\mathbf{x}_i^c, \mathbb{X}_g)$ and, according to Assumption 4, $\mathbf{p}_c(\mathbf{x}_i^u, \mathbf{x}_j^{t,g}) \notin \mathcal{Y}_i^g$. Consequently, from (4.6), $\mathbf{x}_j^{t,g} \notin \mathbb{P}_i^g(\mathcal{Y}_i^g)$. \square

Figure 4.3 illustrates Proposition 14. Two pixels labeled *Ground* (highlighted in orange) are considered. According to (3.54), the pixel \mathbf{p}_1 (highlighted in blue) containing the projection on the CCD array of $\mathbf{x}_j^{t,g}$ cannot be labeled *Ground*. Therefore, $\mathbb{P}_i^g(\mathcal{Y}_i^g)$ (in orange) cannot contain $\mathbf{x}_j^{t,g}$.

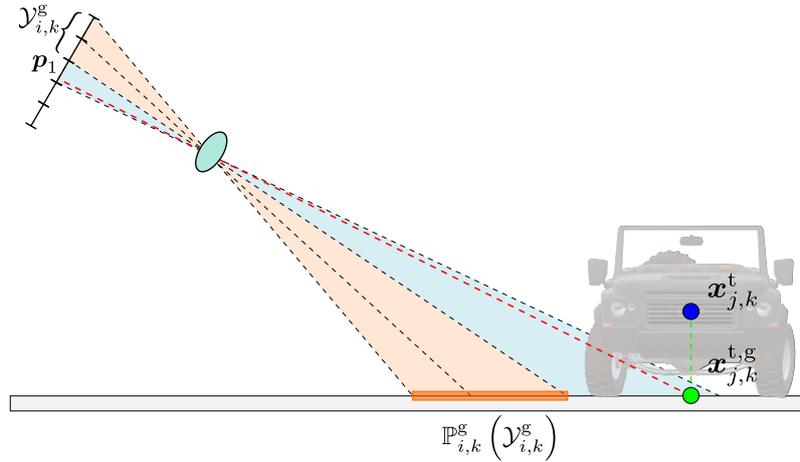


Figure 4.3 – Illustration of the set $\mathbb{P}_i^g(\mathcal{Y}_i^g)$ (in orange) defined as a set which cannot contain the location (in green) $\mathbf{x}_j^{t,g}$, for $j \in \mathcal{N}^t$, of any target

4.3.2 . Using pixels labeled *Obstacle*

This section presents an approach to use pixels labeled *Obstacle* to characterize a subset $\mathbb{X}_{i,k}^o$ of \mathbb{X}_g that cannot contain any target location. The proposed approach relies on Assumption 3.8, stating that, for any target $j \in \mathcal{N}^t$, the distance between $\mathbf{x}_j^{t,g}$ and any obstacle is always larger than some known safety distance r_s^{to} .

For pixels labeled *Obstacle*, Proposition 15 provides a subset of \mathbb{X}_0 intersecting some obstacle \mathbb{S}_m^o .

Proposition 15. If $(n_r, n_c) \in \mathcal{Y}_i^o$, then there exists $m \in \mathcal{N}^o$ such that $\mathbb{P}_i((n_r, n_c)) \cap \mathbb{S}_m^o \neq \emptyset$.

Démonstration. If $(n_r, n_c) \in \mathcal{Y}_i^o$, then, according to Assumption 4, there exists $m \in \mathcal{N}^o$ such that for all $v \in \mathcal{V}_i(n_r, n_c)$, $\rho(\mathbf{x}_i^c, v) = d_v(\mathbf{x}_i^c, \mathbb{S}_m^o)$. Therefore, there exists $v \in \mathcal{V}_i(n_r, n_c)$ and there exists $\mathbf{x} \in \mathbb{S}_m^o \subset \mathbb{X}_0$ such that $d_v(\mathbf{x}_i^c, \{\mathbf{x}\}) = \mathbf{D}_i^o(n_r, n_c)$. This leads to have $d_v(\mathbf{x}_i^c, \{\mathbf{x}\}) \in [\mathbf{D}_i](n_r, n_c)$. In addition, as $(n_r, n_c) \in \mathcal{Y}_i^o \subset \mathcal{Y}_i$, we have $\frac{1}{1+w}\mathbf{D}_i(n_r, n_c) \leq d_{\max}$ and $d_v(\mathbf{x}_i^c, \{\mathbf{x}\}) \leq d_{\max}$ according to (3.36). Therefore $\mathbf{x} \in \mathbb{F}(\mathbf{x}_i^u)$. Consequently, from (4.1), $\mathbf{x} \in \mathbb{P}_i((n_r, n_c))$. So there exists $m \in \mathcal{N}^o$ such that $\mathbb{P}_i((n_r, n_c)) \cap \mathbb{S}_m^o \neq \emptyset$. \square

If, for some $(n_r, n_c) \in \mathcal{Y}_i^o$, $\mathbf{x} \in \mathbb{P}_i((n_r, n_c)) \cap \mathbb{S}_m^o$ would be known, then, according to Assumption 3.8, one would be able to prove that there is no target location in $\mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{to}})$. Nonetheless, as $\mathbb{P}_i((n_r, n_c)) \cap \mathbb{S}_m^o$ is only known to belong to $\mathbb{P}_i((n_r, n_c))$, one considers the set

$$\underline{\mathbb{S}}((n_r, n_c), r_s^{\text{to}}) = \bigcap_{\mathbf{x} \in \mathbf{p}_g(\mathbb{P}_i((n_r, n_c)))} \mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{to}}). \quad (4.7)$$

defined as the intersections of the discs $\mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{to}})$ for all $\mathbf{x} \in \mathbf{p}_g(\mathbb{P}_i((n_r, n_c)))$. According to Proposition 16, $\underline{\mathbb{S}}((n_r, n_c), r_s^{\text{to}})$ is an inner-approximation of the r_s^{to} -ground neighborhood of an obstacle, as illustrated in Figure 4.4.

Proposition 16. *If $(n_r, n_c) \in \mathcal{Y}_i^o$, then there exists $m \in \mathcal{N}^o$, such that*

$$\underline{\mathbb{S}}((n_r, n_c), r_s^{\text{to}}) \subset \mathbb{N}_g(\mathbf{p}_g(\mathbb{S}_m^o), r_s^{\text{to}}). \quad (4.8)$$

Démonstration. If $(n_r, n_c) \in \mathcal{Y}_i^o$, then, according to Proposition 15, there exists $m \in \mathcal{N}^o$ such that $\mathbb{P}_i((n_r, n_c)) \cap \mathbb{S}_m^o \neq \emptyset$. Consequently, $\mathbf{p}_g(\mathbb{P}_i((n_r, n_c))) \cap \mathbf{p}_g(\mathbb{S}_m^o) \neq \emptyset$. Since $(\mathbf{p}_g(\mathbb{P}_i((n_r, n_c))) \cap \mathbf{p}_g(\mathbb{S}_m^o)) \subset \mathbf{p}_g(\mathbb{S}_m^o)$, one has

$$\mathbb{N}_g(\mathbf{p}_g(\mathbb{P}_i((n_r, n_c))) \cap \mathbf{p}_g(\mathbb{S}_m^o), r_s^{\text{to}}) \subset \mathbb{N}_g(\mathbf{p}_g(\mathbb{S}_m^o), r_s^{\text{to}}). \quad (4.9)$$

Moreover, one has

$$\bigcap_{\mathbf{x} \in \mathbf{p}_g(\mathbb{P}_i((n_r, n_c)))} \mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{to}}) \subset \bigcap_{\mathbf{x} \in \mathbf{p}_g(\mathbb{P}_i((n_r, n_c))) \cap \mathbf{p}_g(\mathbb{S}_m^o)} \mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{to}}), \quad (4.10)$$

as well as

$$\bigcap_{\mathbf{x} \in \mathbf{p}_g(\mathbb{P}_i((n_r, n_c))) \cap \mathbf{p}_g(\mathbb{S}_m^o)} \mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{to}}) \subset \mathbb{N}_g(\mathbf{p}_g(\mathbb{P}_i((n_r, n_c))) \cap \mathbf{p}_g(\mathbb{S}_m^o), r_s^{\text{to}}). \quad (4.11)$$

Finally, one gets $\underline{\mathbb{S}}((n_r, n_c), r_s^{\text{to}}) \subset \mathbb{N}_g(\mathbf{p}_g(\mathbb{S}_m^o), r_s^{\text{to}})$. \square

Consequently, an inner-approximation of the r_s^{to} -ground neighborhood of all the obstacles located within the FoV of UAV i can be obtained by characterizing the union of all $\underline{\mathbb{S}}((n_r, n_c), r_s^{\text{to}})$ over all pixels $(n_r, n_c) \in \mathcal{Y}_i^o$. Therefore, UAV i is able to characterize a set estimate

$$\underline{\mathbb{X}}_i^o = \bigcup_{(n_r, n_c) \in \mathcal{Y}_i^o} \underline{\mathbb{S}}((n_r, n_c), r_s^{\text{to}}) \quad (4.12)$$

which cannot contain any target location as stated by the following proposition.

Proposition 17. *If $\mathcal{Y}_i^o \neq \emptyset$, then the set $\underline{\mathbb{X}}_i^o$ is such that $\mathbf{x}_j^{\text{tg}} \notin \underline{\mathbb{X}}_i^o$, for all $j \in \mathcal{N}^t$.*

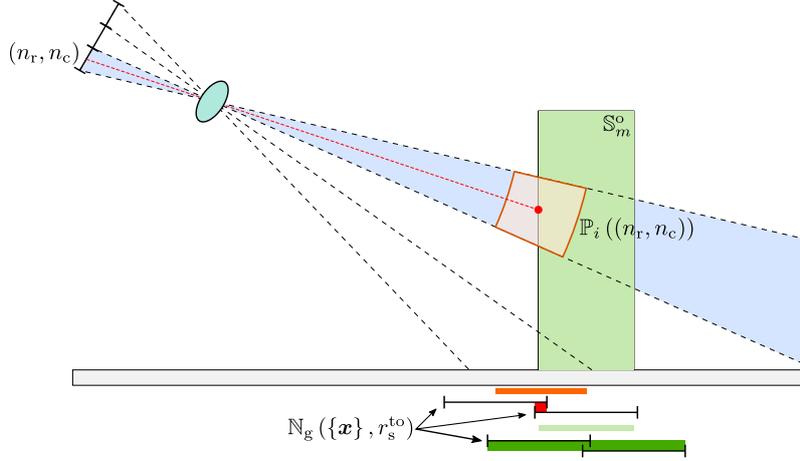


Figure 4.4 – Illustration of Proposition 16 : The obstacle \mathbb{S}_m^o and the projection on the ground $\mathbf{p}_g(\mathbb{S}_m^o)$ of its shape \mathbb{S}_m^o are represented by the light-green sets. The set $\mathbb{P}_i((n_r, n_c))$ associated to a pixel (n_r, n_c) labeled *Obstacle* and its ground projection are represented by the orange sets. $\mathbb{N}_g(\{\mathbf{x}\}, r_s^{to})$ is represented by the black segments. The red set is $\underline{\mathbb{S}}((n_r, n_c), r_s^{to})$ and the dark-green set is $\mathbb{N}_g(\mathbf{p}_g(\mathbb{S}_m^o), r_s^{to})$.

Démonstration. Using Proposition 16, for each $(n_r, n_c) \in \mathcal{Y}_i^o$, there exists $m \in \mathcal{N}^o$ such that $\underline{\mathbb{S}}((n_r, n_c), r_s^{to}) \subset \mathbb{N}_g(\mathbf{p}_g(\mathbb{S}_m^o), r_s^{to})$. Then

$$\bigcup_{(n_r, n_c) \in \mathcal{Y}_i^o} \underline{\mathbb{S}}((n_r, n_c), r_s^{to}) \subset \bigcup_{m \in \mathcal{N}^o} \mathbb{N}_g(\mathbf{p}_g(\mathbb{S}_m^o), r_s^{to}). \quad (4.13)$$

Consequently $\underline{\mathbb{X}}_i^o \subset \bigcup_{m \in \mathcal{N}^o} \mathbb{N}_g(\mathbf{p}_g(\mathbb{S}_m^o), r_s^{to})$. According to Assumption 3.8, we have

$$\forall j \in \mathcal{N}^t, \forall m \in \mathcal{N}^o, \mathbf{x}_j^{tg} \notin \mathbb{N}_g(\mathbf{p}_g(\mathbb{S}_m^o), r_s^{to}) \quad (4.14)$$

$$\Leftrightarrow \forall j \in \mathcal{N}^t, \mathbf{x}_j^{tg} \notin \bigcup_{m \in \mathcal{N}^o} \mathbb{N}_g(\mathbf{p}_g(\mathbb{S}_m^o), r_s^{to}) \quad (4.15)$$

therefore

$$\forall j \in \mathcal{N}^t, \mathbf{x}_j^{tg} \notin \underline{\mathbb{X}}_i^o. \quad (4.16)$$

□

4.3.3 . Using pixels labeled as Target

This section presents a method to characterize a subset $\underline{\mathbb{X}}_{i,j}^{t,m}$ of $\underline{\mathbb{X}}_g$ that cannot contain the location of any target $\ell \in \mathcal{N}^t \setminus \{j\}$ using the information associated with the identified target $j \in \mathcal{D}_i^t$.

If target j is identified at time t_k , then, with the approach presented in Section 4.2, a set estimate $\underline{\mathbb{X}}_{i,j}^{t,m}$ containing its location is available. Using Assumption (3.11), we have the following proposition.

Proposition 18. *If $j \in \mathcal{D}_i^t$ then the set estimate*

$$\underline{\mathbb{X}}_{i,j}^{t,m} = \bigcap_{\mathbf{x} \in \underline{\mathbb{X}}_{i,j}^{t,m}} \mathbb{N}_g(\{\mathbf{x}\}, r_s^{tt}) \quad (4.17)$$

is such that for all $\ell \in \mathcal{N}^t \setminus \{j\}$, $\mathbf{x}_\ell^{\text{tg}} \notin \underline{\mathbb{X}}_{i,j}^{\text{t},m}$.

Démonstration. If $j \in \mathcal{D}_i^t$, then according to Proposition 13, the set estimate $\underline{\mathbb{X}}_{i,j}^{\text{t},m}$ is such that $\mathbf{x}_j^{\text{tg}} \in \underline{\mathbb{X}}_{i,j}^{\text{t},m}$. Consequently, one has

$$\bigcap_{\mathbf{x} \in \underline{\mathbb{X}}_{i,j}^{\text{t},m}} \mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{tt}}) \subset \mathbb{N}_g(\{\mathbf{x}_j^{\text{tg}}\}, r_s^{\text{tt}}). \quad (4.18)$$

Therefore, according to Assumption (3.11), $\mathbf{x}_\ell^{\text{tg}} \notin \underline{\mathbb{X}}_{i,j}^{\text{t},m}$ for all $\ell \in \mathcal{N}^t \setminus \{j\}$. \square

4.4 . Estimation of the hidden area

Some parts of $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$ may not be seen by UAV i when they are hidden by an obstacle, a target, a UAV, or something that has not been identified.

Consider, for now, a point $\mathbf{x} \in \mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$ that is hidden by a detected obstacle, *i.e.*, there exists a $(n_r, n_c) \in \mathcal{Y}_i^o$ such that $\mathbf{p}_c(\mathbf{x}_i^u, \mathbf{x}) \in (n_r, n_c)$. If this point on the ground has been hidden by an obstacle, then, there exists an obstacle m , with $m \in \mathcal{N}^o$ such that $d_v(\mathbf{x}_i^c, \mathbb{S}_m^o) < d_v(\mathbf{x}_i^c, \{\mathbf{x}\})$, where $\mathbf{v} \in \mathcal{V}_i(n_r, n_c)$ is collinear with the segment $[\mathbf{x}_i^c, \mathbf{x}]$. Thus, according to (3.31), we have that $\rho(\mathbf{x}_i^c, \mathbf{v}) < d_v(\mathbf{x}_i^c, \{\mathbf{x}\})$. The set of points belonging to $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$ that are hidden by a detected obstacle is then

$$\begin{aligned} \underline{\mathbb{H}}_i^o &= \{ \mathbf{x} \in \mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g \mid \\ &\quad \exists (n_r, n_c) \in \mathcal{Y}_i^o, \exists \mathbf{v} \in \mathcal{V}_i(n_r, n_c), \rho(\mathbf{x}_i^c, \mathbf{v}) < d_v(\mathbf{x}_i^c, \{\mathbf{x}\}) \}. \end{aligned} \quad (4.19)$$

The same sets can be defined for the points belonging to $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$ that are hidden either by a detected target or by something not labeled, *i.e.*,

$$\begin{aligned} \underline{\mathbb{H}}_i^t &= \{ \mathbf{x} \in \mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g \mid \\ &\quad \exists (n_r, n_c) \in \mathcal{Y}_i^t, \exists \mathbf{v} \in \mathcal{V}_i(n_r, n_c), \rho(\mathbf{x}_i^c, \mathbf{v}) < d_v(\mathbf{x}_i^c, \{\mathbf{x}\}) \}. \end{aligned} \quad (4.20)$$

$$\begin{aligned} \underline{\mathbb{H}}_i^n &= \{ \mathbf{x} \in \mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g \mid \\ &\quad \exists (n_r, n_c) \in \mathcal{Y}_i^n, \exists \mathbf{v} \in \mathcal{V}_i(n_r, n_c), \rho(\mathbf{x}_i^c, \mathbf{v}) < d_v(\mathbf{x}_i^c, \{\mathbf{x}\}) \}. \end{aligned} \quad (4.21)$$

Consequently, the set

$$\underline{\mathbb{H}}_i^{\text{CVS}} = \underline{\mathbb{H}}_i^o \cup \underline{\mathbb{H}}_i^t \cup \underline{\mathbb{H}}_i^n, \quad (4.22)$$

represents the portion of the ground that UAV i cannot see. The main difficulty to characterize (4.22) comes from the fact that $\rho(\mathbf{x}_i^c, \mathbf{v})$ cannot be evaluated without a perfect knowledge of \mathbb{S}^{ROl} . Nevertheless, UAV i is able to outer-approximate $\underline{\mathbb{H}}_i^{\text{CVS}}$ by a set estimate $\underline{\mathbb{H}}_i^{\text{CVS}}$ that will be useful in the UAV exploration process in Chapter 5.

To get this outer-approximation of $\underline{\mathbb{H}}_i^{\text{CVS}}$, we assume that as soon as a pixel (n_r, n_c) is labeled *Obstacle*, *Target*, or *Unknown*, *i.e.*, $(n_r, n_c) \in \mathcal{Y}_i^o \cup \mathcal{Y}_i^t \cup \mathcal{Y}_i^n$, then it is not illuminated by the ground. According to Assumption 4, this assumption is true for pixels labeled *Obstacle* and *Target*, but not necessarily for pixels labeled *Unknown*. Consequently, we can prove that $\underline{\mathbb{H}}_i^o \subseteq \mathbb{P}_i^g(\mathcal{Y}_i^o)$, and $\underline{\mathbb{H}}_i^t \subseteq$

$\mathbb{P}_i^g(\mathcal{Y}_i^t)$ since, according to Assumption 4, for all $\mathbf{v} \in \mathcal{V}_i(n_r, n_c)$, with $(n_r, n_c) \in \mathcal{Y}_i^o \cup \mathcal{Y}_i^t$, one has $\rho(\mathbf{x}_i^c, \mathbf{v}) \leq d_{\mathbf{v}}(\mathbf{x}_i^c, \mathbb{X}_g)$. We have an equality only when the point $\mathbf{x} \in \mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$ belongs to the footprint of an obstacle $m \in \mathcal{N}^o$, i.e., $\mathbf{x} \in \mathbb{S}_m^o \cap \mathbb{X}_g$, or if it belongs to $\mathbb{S}^t(\mathbf{x}_j^t) \cap \mathbb{X}_g$ for a target $j \in \mathcal{N}^t$. The same can be done with pixel labeled *Unknown*, i.e., $\mathbb{H}_i^n \subset \mathbb{P}_i^g(\mathcal{Y}_i^n)$.

Consequently, an outer-approximation of $\mathbb{H}_i^{\text{CVS}}$ is

$$\mathbb{H}_i^{\text{CVS}} = \mathbb{P}_i^g(\mathcal{Y}_i^o \cup \mathcal{Y}_i^t \cup \mathcal{Y}_i^n). \quad (4.23)$$

4.5 . Estimation algorithm

This section describes an adaptation of the distributed set-membership target location estimator introduced in (Ibenthal et al., 2021) to exploit measurements provided by the CVS. Just before time t_k , UAV i has access to the possible locations of identified targets, i.e., $\mathbb{X}_{i,j,k-1}^t$ for each $j \in \mathcal{L}_{i,k-1}^t$, and the possible location of the targets still to be identified, i.e., $\overline{\mathbb{X}}_{i,k-1}^t$. The UAV has also access to a subset of \mathbb{X}_g that represents an inner-approximation of the $r_s^{t_0}$ -ground neighborhood of all the previously detected obstacles, i.e., $\mathbb{X}_{i,k-1}^o$. At time t_0 , the set estimates of UAV i are initialized as $\mathcal{L}_{i,0}^t = \emptyset$, $\mathcal{X}_{i,0}^t = \emptyset$, $\overline{\mathbb{X}}_{i,0}^t = \mathbb{X}_g$, and $\mathbb{X}_{i,0}^o = \emptyset$.

4.5.1 . Prediction step

The predicted set $\mathbb{X}_{i,j,k|k-1}^t$ of possible future possible locations of the previously identified target $j \in \mathcal{L}_{i,k-1}^t$ is evaluated from $\mathbb{X}_{i,j,k-1}^t$, the target dynamics (3.2), and the known bounds $[\mathbf{v}^t]$ of its control input

$$\mathbb{X}_{i,j,k|k-1}^t = \{ \mathbf{f}^t(\mathbf{x}, \mathbf{v}) \in \mathbb{X}_g \mid \mathbf{x} \in \mathbb{X}_{i,j,k-1}^t, \mathbf{v} \in [\mathbf{v}^t] \} \quad (4.24)$$

$$= \mathbf{f}^t(\mathbb{X}_{i,j,k-1}^t, [\mathbf{v}^t]) \cap \mathbb{X}_g. \quad (4.25)$$

Similarly, the predicted set $\overline{\mathbb{X}}_{i,k|k-1}^t$ of all possible locations of targets still to be identified is

$$\overline{\mathbb{X}}_{i,k|k-1}^t = \mathbf{f}^t(\overline{\mathbb{X}}_{i,k-1}^t, [\mathbf{v}^t]) \cap \mathbb{X}_g. \quad (4.26)$$

Obstacles being static, the predicted set $\mathbb{X}_{i,k|k-1}^o$ is

$$\mathbb{X}_{i,k|k-1}^o = \mathbb{X}_{i,k-1}^o. \quad (4.27)$$

4.5.2 . Update using CVS measurements

The CVS of UAV i provides $\mathbf{I}_{i,k}$, $\mathbf{L}_{i,k}$, $\mathbf{D}_{i,k}$, $\mathcal{D}_{i,k}^t$, and $\mathcal{B}_{i,k}^t$. From these CVS measurements, UAV i has characterized several set estimates. Some of them are guaranteed to contain the location of an identified target. Others are guaranteed to contain no target location. Using these set estimates, UAV i updates the knowledge it has about the environment and the possible locations of both identified and unidentified targets.

Using Proposition 17, the list of pixels labeled *Obstacle* $\mathcal{Y}_{i,k}^o$ is used to characterize the set $\underline{\mathbb{X}}_{i,k}^o$ that is guaranteed to contain no target location. This set represents an inner-approximation of the

r_s^{to} -ground neighborhoods of all the obstacles that are detected at time t_k . Thus, the update of the region around the obstacles that cannot contain any target location is,

$$\mathbb{X}_{i,k|k}^{\text{o}} = \mathbb{X}_{i,k|k-1}^{\text{o}} \cup \underline{\mathbb{X}}_{i,k}^{\text{o}}. \quad (4.28)$$

Regarding the set $\overline{\mathbb{X}}_{i,k|k-1}^{\text{t}}$ which may contain unidentified targets, Propositions 14 and 17 showed that $\mathbb{P}_{i,k}^{\text{g}}(\mathcal{Y}_{i,k}^{\text{g}})$ and $\underline{\mathbb{X}}_{i,k}^{\text{o}}$ (and therefore $\mathbb{X}_{i,k|k}^{\text{o}}$) cannot contain the location of any target. In addition, since $\overline{\mathbb{X}}_{i,k|k-1}^{\text{t}}$ should contain only the locations of potential unidentified target, thus, according to Proposition 18, the set $\underline{\mathbb{X}}_{i,j,k}^{\text{t,m}}$ for all targets $j \in \mathcal{D}_{i,k}^{\text{t}}$ cannot contain the location of an unidentified target. Consequently, the set $\overline{\mathbb{X}}_{i,k|k-1}^{\text{t}}$ is updated to get

$$\overline{\mathbb{X}}_{i,k|k}^{\text{t}} = \overline{\mathbb{X}}_{i,k|k-1}^{\text{t}} \setminus \left(\mathbb{P}_{i,k}^{\text{g}}(\mathcal{Y}_{i,k}^{\text{g}}) \cup \mathbb{X}_{i,k|k}^{\text{o}} \cup \bigcup_{\ell \in \mathcal{D}_{i,k}^{\text{t}}} \underline{\mathbb{X}}_{i,\ell,k}^{\text{t,m}} \right), \quad (4.29)$$

Then, for the target set estimate $\mathbb{X}_{i,j,k|k-1}^{\text{t}}$, we know that $\mathbb{P}_{i,k}^{\text{g}}(\mathcal{Y}_{i,k}^{\text{g}}) \cup \mathbb{X}_{i,k|k}^{\text{o}}$ cannot contain any target. Furthermore, according to Proposition 18, for any target $j \in \mathcal{N}^{\text{t}}$, its location cannot be contained in the union of the $\underline{\mathbb{X}}_{i,\ell,k}^{\text{t,m}}$ over all identified targets except itself, *i.e.*, $\mathbf{x}_{j,k}^{\text{t,g}} \notin \bigcup_{\ell \in \mathcal{D}_{i,k}^{\text{t}} \setminus \{j\}} \underline{\mathbb{X}}_{i,\ell,k}^{\text{t,m}}$. Therefore, for each previously identified target $j \in \mathcal{L}_{i,k-1}^{\text{t}} \setminus \mathcal{D}_{i,k}^{\text{t}}$ which is not identified at time t_k , the set estimate $\mathbb{X}_{i,j,k|k-1}^{\text{t}}$ is updated to get

$$\mathbb{X}_{i,j,k|k}^{\text{t}} = \mathbb{X}_{i,j,k|k-1}^{\text{t}} \setminus \left(\mathbb{P}_{i,k}^{\text{g}}(\mathcal{Y}_{i,k}^{\text{g}}) \cup \mathbb{X}_{i,k|k}^{\text{o}} \cup \bigcup_{\ell \in \mathcal{D}_{i,k}^{\text{t}} \setminus \{j\}} \underline{\mathbb{X}}_{i,\ell,k}^{\text{t,m}} \right). \quad (4.30)$$

For each previously identified target j which has been identified at time t_k , *i.e.*, $j \in \mathcal{L}_{i,k-1}^{\text{t}} \cap \mathcal{D}_{i,k}^{\text{t}}$, then, Proposition 13 showed that $\mathbb{X}_{i,j,k}^{\text{t,m}}$ is guaranteed to contain $\mathbf{x}_{j,k}^{\text{t,g}}$. Therefore, the update of the set $\mathbb{X}_{i,j,k|k-1}^{\text{t}}$ containing all of its predicted possible locations is

$$\mathbb{X}_{i,j,k|k}^{\text{t}} = \left(\mathbb{X}_{i,j,k|k-1}^{\text{t}} \cap \mathbb{X}_{i,j,k}^{\text{t,m}} \right) \setminus \left(\mathbb{P}_{i,k}^{\text{g}}(\mathcal{Y}_{i,k}^{\text{g}}) \cup \mathbb{X}_{i,k|k}^{\text{o}} \cup \bigcup_{\ell \in \mathcal{D}_{i,k}^{\text{t}} \setminus \{j\}} \underline{\mathbb{X}}_{i,\ell,k}^{\text{t,m}} \right). \quad (4.31)$$

In the case of a newly identified target $j \in \mathcal{D}_{i,k}^{\text{t}} \setminus \mathcal{L}_{i,k-1}^{\text{t}}$, its possible locations known to UAV i at time t_k are only known to belong to $\overline{\mathbb{X}}_{i,k|k-1}^{\text{t}}$. Consequently, UAV i characterizes the set estimate

$$\mathbb{X}_{i,j,k|k}^{\text{t}} = \left(\overline{\mathbb{X}}_{i,k|k-1}^{\text{t}} \cap \mathbb{X}_{i,j,k}^{\text{t,m}} \right) \setminus \left(\mathbb{P}_{i,k}^{\text{g}}(\mathcal{Y}_{i,k}^{\text{g}}) \cup \mathbb{X}_{i,k|k}^{\text{o}} \cup \bigcup_{\ell \in \mathcal{D}_{i,k}^{\text{t}} \setminus \{j\}} \underline{\mathbb{X}}_{i,\ell,k}^{\text{t,m}} \right). \quad (4.32)$$

Finally, the list of indexes of identified targets becomes

$$\mathcal{L}_{i,k|k}^{\text{t}} = \mathcal{L}_{i,k-1}^{\text{t}} \cup \mathcal{D}_{i,k}^{\text{t}}. \quad (4.33)$$

4.5.3 . Update after communication with neighbors

Once UAV i has updated its knowledge using the CVS measurements it has acquired at time t_k , it exchanges information with its neighbors. This data exchange consists of broadcasting to the neighbors the sets $\mathbb{X}_{i,k|k}^o$, $\bar{\mathbb{X}}_{i,k|k}^t$, $\mathbb{X}_{i,j,k|k}^t$ for each $j \in \mathcal{L}_{i,k|k}^t$ and the list $\mathcal{L}_{i,k|k}^t$, while receiving theirs.

Each UAV ℓ , with $\ell \in \mathcal{N}^u$, broadcasts the list of indexes of identified targets $\mathcal{L}_{\ell,k|k}^t$, the r_s^{to} -ground neighborhood of obstacles $\mathbb{X}_{\ell,k|k}^o$, the set estimates $\mathbb{X}_{\ell,j,k|k}^t$, $j \in \mathcal{L}_{\ell,k|k}^t$ and the set of possible locations of targets still to be identified $\bar{\mathbb{X}}_{\ell,k|k}^t$. UAV i receives this information from its neighbors with indexes in $\mathcal{N}_{i,k}$. We recall that we always have $i \in \mathcal{N}_{i,k}$. As indicated in Section 3.3.1, communication is assumed perfect and without delay. More practical aspects related to the scheduling of communications are detailed in Section 5.3.2.

Using the information received from its neighbors, UAV i updates the list of identified targets as

$$\mathcal{L}_{i,k}^t = \bigcup_{\ell \in \mathcal{N}_{i,k}} \mathcal{L}_{\ell,k|k}^t, \quad (4.34)$$

the r_s^{to} -neighborhood of obstacles as

$$\mathbb{X}_{i,k}^o = \bigcup_{\ell \in \mathcal{N}_{i,k}} \mathbb{X}_{\ell,k|k}^o, \quad (4.35)$$

and the set of possible locations of targets still to be identified as

$$\bar{\mathbb{X}}_{i,k}^t = \bigcap_{\ell \in \mathcal{N}_{i,k}} \bar{\mathbb{X}}_{\ell,k|k}^t. \quad (4.36)$$

Then, for each target $j \in \mathcal{L}_{i,k}^t$, one has

$$\mathbb{X}_{i,j,k}^t = \left(\bigcap_{\substack{\ell \in \mathcal{N}_{i,k} \\ \text{st } j \in \mathcal{L}_{\ell,k|k}^t}} \mathbb{X}_{\ell,j,k|k}^t \right) \cap \left(\bigcap_{\substack{\ell \in \mathcal{N}_{i,k} \\ \text{st } j \notin \mathcal{L}_{\ell,k|k}^t}} \bar{\mathbb{X}}_{\ell,k|k}^t \right). \quad (4.37)$$

The expression (4.37) accounts for the estimate $\mathbb{X}_{\ell,j,k|k}^t$ of the location of target j made by the neighbor UAVs which have already identified target j . It also accounts for the fact that $\mathbf{x}_{j,k}^t \in \bar{\mathbb{X}}_{\ell,k|k}^t$ for all neighbors of UAV i which have not identified target j up to time t_k .

4.6 . Update of the occupancy-elevation map

This section presents an approach to build a map from the CVS measurements available to UAV i . Section 4.3.2 explains how to obtain an inner-approximation of the r_s^{to} -ground neighborhood of the detected obstacles, *i.e.*, $\underline{\mathbb{X}}_{i,k}^o \subset \mathbb{N}_g(\mathbf{p}_g(\mathbb{S}_m^o), r_s^{\text{to}})$. Nonetheless, the map is used to get a rough approximation of the location and the 3D shape of the obstacles within the RoI. Thus, $\underline{\mathbb{X}}_{i,k}^o$ will not be used since it provides no information about the height of the obstacles. Additionally, the map is **not** used to

estimate the location of the targets. Its main purpose is to improve the update of the UAV trajectories, as presented in Chapter 5, by predicting $\mathbb{H}_{i,k+\tau}^{\text{CVS}}$, with $\tau \geq 1$.

The proposed approach relies on an Occupancy-Elevation Map (OEM) (Souza and Goncalves, 2016; Du et al., 2018) introduced in Section 4.6.1. Then, Section 4.6.2 explains the update of the OEM for each UAV using the CVS measurements. Finally, Section 4.6.3 describes its update with the information broadcast by the neighbors of each UAV.

4.6.1 . Definition of an occupancy-elevation map

The combination of a 2D-occupancy map with an elevation map results in a 2.5D representation to map a 3D environment. With this approach, obstacles spread within the RoI are approximated by a list of parallelepipeds, see Figure 4.5.

The OEM $\mathcal{M}_{i,k}$ managed by UAV i at time t_k is a 2D grid composed of non-overlapping square cells of same dimension. Each cell is uniquely identified with indexes η in the set \mathcal{N}^M . A cell $\eta \in \mathcal{N}^M$, is characterized by the location of its center $\mathbf{c}(\eta) \in \mathbb{X}_g$, its status $s_{i,k}(\eta)$, and its elevation $h_{i,k}(\eta)$. The status $s_{i,k}(\eta)$ takes its value in the set $\{U, O, E\}$ where U stands for *unexplored*, O for *occupied*, and E for *empty*. Then, $\mathcal{M}_{i,k} = \{\mathbf{c}(\eta), s_{i,k}(\eta), h_{i,k}(\eta)\}_{\eta \in \mathcal{N}^M}$. An *unexplored* cell represents an area where no information has been collected yet. An *occupied* cell represents an area where an obstacle has been detected. An *empty* cell represents an area where there is no obstacle, *i.e.*, only the ground has been observed.

Initially, $s_{i,0}(\eta) = \{U\}$ and $h_{i,0}(\eta) = 0$ for all $\eta \in \mathcal{N}^M$.

Figure 4.5 provides an example of OEM where some cells are *empty* (in green), others contain an obstacle with a measured height (in red), and the remaining cells are still *unexplored* (in blue).

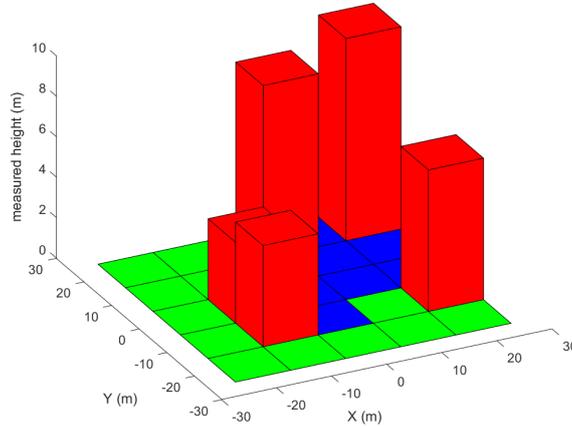


Figure 4.5 – Representation of an occupancy-elevation map $\mathcal{M}_{i,k}$. Green cells represent *empty* cells; the portions of \mathbb{X}_g covered by those cells are considered free of obstacles. Blue cells are *unexplored* cells; no useful information related to the map topology is available. The red parallelepipeds are obtained from the location and size of the *occupied* cells (an obstacle has been detected on those cells) and the measured height (related to those detected obstacles).

4.6.2 . Update using CVS information

Just before time t_k , assume that UAV i has access to the OEM $\mathcal{M}_{i,k-1} = \{\mathbf{c}(\eta), s_{i,k-1}(\eta), h_{i,k-1}(\eta)\}_{\eta \in \mathcal{N}^M}$. Obstacles are static. Consequently, no prediction of the evolution of the OEM is required. Thus, $h_{i,k|k}(\eta) = h_{i,k-1}(\eta)$ and $s_{i,k|k}(\eta) = s_{i,k-1}(\eta)$ for all $\eta \in \mathcal{N}^M$.

At time t_k new measurements are provided by the CVS of UAV i . They are used to get an updated version $\mathcal{M}_{i,k|k}$ of the OEM.

Detection and approximation of an obstacle

Consider a pixel labeled Obstacle $(n_r, n_c) \in \mathcal{Y}_{i,k}^o$. According to Proposition 15, then there exists $m \in \mathcal{N}^o$ such that $\mathbb{P}_{i,k}((n_r, n_c)) \cap \mathbb{S}_m^o \neq \emptyset$. Therefore, $\mathbb{P}_{i,k}((n_r, n_c))$ has a non-empty intersection with at least one obstacle. Nonetheless, points belonging to this intersection, and consequently to \mathbb{S}_m^o cannot be easily characterized. This is why only an approximation of the obstacles is provided by the OEM. Then, to characterize the approximation of the shape of the obstacle $m \in \mathcal{N}^o$ that has illuminated pixel (n_r, n_c) , we consider the following point of $\mathbb{P}_{i,k}((n_r, n_c))$

$$\bar{\mathbf{x}}(n_r, n_c) = \mathbf{x}_{i,k}^c + \frac{\mathbf{D}_{i,k}(n_r, n_c)}{1 + w} \mathbf{v}^c(n_c, n_r), \quad (4.38)$$

where $\mathbf{v}^c(n_c, n_r)$ is the unit vector belonging to $\mathcal{V}_{i,k}(n_r, n_c)$ which defines the light-ray illuminating the center of pixel (n_r, n_c) . The point $\bar{\mathbf{x}}(n_r, n_c)$ of $\mathbb{P}_{i,k}((n_r, n_c))$ is the farthest from $\mathbf{x}_{i,k}^c$ that may have illuminated the center of the pixel (n_r, n_c) . Since the considered obstacles in our structured environment are likely to be buildings, $\bar{\mathbf{x}}(n_r, n_c)$ is likely to be inside an obstacle $m \in \mathcal{N}^o$. Consequently, if the projection on \mathbb{X}_g of $\bar{\mathbf{x}}(n_r, n_c)$, for a $(n_r, n_c) \in \mathcal{Y}_{i,k}^o$, belongs to a cell that is not *empty*, then that cell is updated as *occupied*. Moreover, we choose $h_{\max}(\mathbb{P}_{i,k}((n_r, n_c)))$ as the estimated height of the detected obstacle, where $h_{\max}(\mathbb{P})$ returns the highest third coordinate value over all the elements contained in \mathbb{P} . Thus, one has for all $(n_r, n_c) \in \mathcal{Y}_{i,k}^o$

$$\begin{aligned} & \text{if } \exists \eta \in \mathcal{N}^M \text{ such that } \mathbf{p}_g(\bar{\mathbf{x}}(n_r, n_c)) \in \text{cell}(\eta) \\ & \text{and } s_{i,k|k}(\eta) \in \{\mathbf{U}, \mathbf{O}\}, \\ & \text{then } s_{i,k|k}(\eta) = \{\mathbf{O}\}, \\ & \text{and } h_{i,k|k}(\eta) = \max(h_{i,k|k}(\eta), h_{\max}(\mathbb{P}_{i,k}((n_r, n_c)))) , \end{aligned} \quad (4.39)$$

where $\text{cell}(\eta)$ is the subset of \mathbb{X}_g that is included in that cell. Figure 4.6 illustrates this update step.

Detection and approximation of the ground

For all pixels (n_r, n_c) labeled *Ground*, i.e., for all $(n_r, n_c) \in \mathcal{Y}_{i,k}^g$, the state of cells whose center belongs to $\mathbb{P}_{i,k}^g((n_r, n_c))$ is updated to *empty* and their height is set to 0. Consequently, one has for all $(n_r, n_c) \in \mathcal{Y}_{i,k}^g$

$$\text{if } \mathbf{c}(n) \in \mathbb{P}_{i,k}^g((n_r, n_c)) \text{ then } s_{i,k|k}(\eta) = \{\mathbf{E}\}, h_{i,k|k}(\eta) = 0. \quad (4.40)$$

Figure 4.6 illustrates this update.

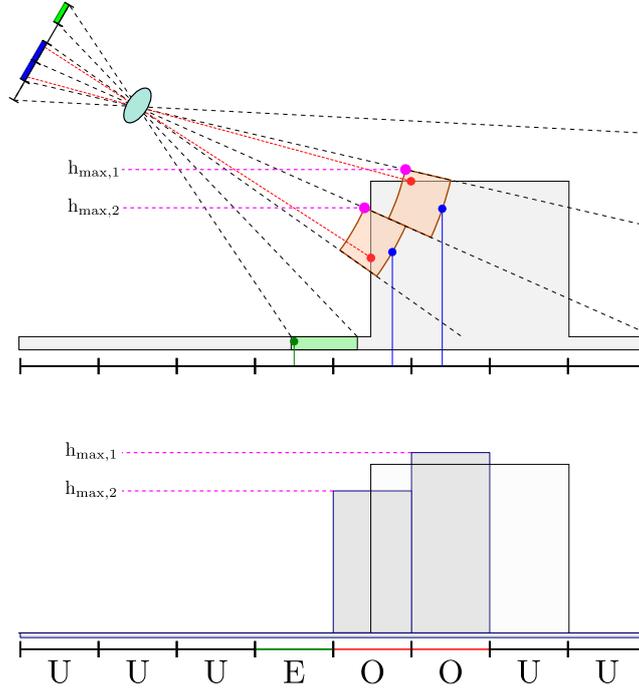


Figure 4.6 – Illustration of the update of the OEM. The blue pixels are labeled *Obstacle*. From those pixels, UAV i characterizes their $\mathbb{P}_{i,k}(\cdot)$ (in orange). The blue and purple dots are their \bar{x} and h_{\max} . The green pixel is labeled *Ground*. The green set is the $\mathbb{P}_{i,k}^g(\cdot)$ related to this pixel and the green dot is the center $c(\eta)$ where the cell η is the only cell that has its center contained in $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$. Assuming that each cell was previously unexplored, the fifth and sixth cells are updated as *occupied* with the correct elevation. The fourth cell is updated as *empty*.

4.6.3 . Update after communication with neighbors

UAV i also updates its OEM as follows.

A cell considered as *empty* by at least one of the UAVs is updated to *empty*. Thus, for all $\eta \in \mathcal{N}^M$, if there exists $\ell \in \mathcal{N}_{i,k}$, such that $s_{\ell,k|k}(\eta) = \{E\}$, then $s_{i,k}(\eta) = \{E\}$ and $h_{i,k}(\eta) = 0$.

After this update, for a cell that is not *empty*, if that cell is considered *occupied* by at least one of the UAVs, then it is updated to *occupied* and the height is set to the maximum observed height. Thus, for all $\eta \in \mathcal{N}^M$, such that there does not exist any $\ell \in \mathcal{N}_{i,k}$ with $s_{\ell,k|k}(\eta) = \{E\}$, but such that there exists $\ell' \in \mathcal{N}_{i,k}$ with $s_{\ell',k|k}(\eta) = \{O\}$, then $s_{i,k}(\eta) = \{O\}$ and $h_{i,k}(\eta) = \max_{\ell \in \mathcal{N}_{i,k}} (h_{\ell,k|k}(\eta))$.

4.6.4 . Approximation of $\mathbb{H}_{i,k}^{\text{CVS}}$ using the OEM

This section presents an approach to approximate $\mathbb{H}_{i,k}^{\text{CVS}}$ by using the OEM.

At time t_k , the OEM maintained by UAV i can be used to define the set of boxes

$$\{\mathbb{B}_{i,k}^M(\eta) \mid s_{i,k}(\eta) = \{O\}, \eta \in \mathcal{N}^M\}. \quad (4.41)$$

This set is an approximation of the shape of all obstacles that have been detected up to time t_k . For each $\mathbb{B}_{i,k}^M(\eta)$, using the approach developed in (Reboul et al., 2019), UAV i , with state \mathbf{x}_i^u , is able to

characterize a set $\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_i^u, \eta) \subset \mathbb{X}_g$ representing the portion of the ground occluded by $\mathbb{B}_{i,k}^M(\eta)$, see Figure 4.7.

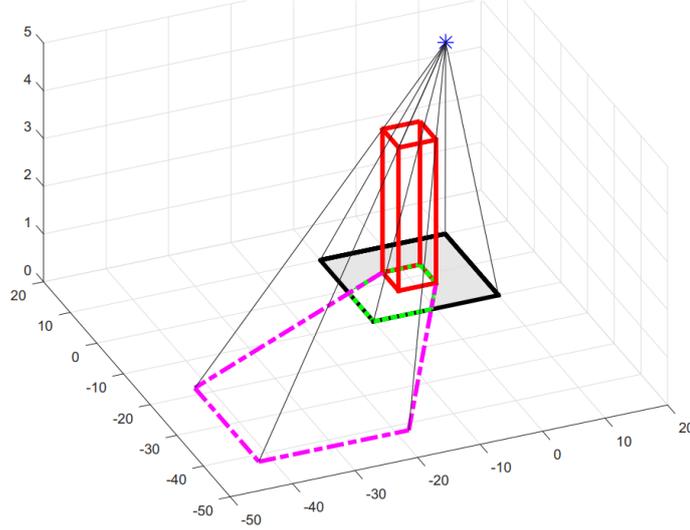


Figure 4.7 – Illustration of the portion of the ground occluded by an obstacle approximated by the red box $\mathbb{B}_{i,k}^M(\eta)$. The UAV location is represented by the blue star, the set $\mathbb{F}(\mathbf{x}_{i,k}^u) \cap \mathbb{X}_g$ is delimited in black, and the set $\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_{i,k}^u, \eta)$ is delimited in purple. Thus, the portion of $\mathbb{F}(\mathbf{x}_{i,k}^u)$ that is occluded by $\mathbb{B}_{i,k}^M(\eta)$ (the set delimited in green) is the intersection between $\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_{i,k}^u, \eta)$ and $\mathbb{F}(\mathbf{x}_{i,k}^u)$. Figure taken from (Reboul et al., 2019).

To characterize $\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_i^u, \eta)$ for a UAV with state \mathbf{x}_i^u , one has to account for the locations of the optical center \mathbf{x}_i^c of the camera and of the 8 vertices of the box $\mathbb{B}_{i,k}^M(\eta)$. Since the box $\mathbb{B}_{i,k}^M(\eta)$ is convex, the portion of the ground hidden by that box is convex too. Then, we evaluate the convex hull of the intersection of \mathbb{X}_g with the 8 lines passing through \mathbf{x}_i^c and one of the 8 vertices of $\mathbb{B}_{i,k}^M(\eta)$.

Then the portion of the ground occluded by all boxes within (4.41) is

$$\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_i^u) = \bigcup_{\eta \in \mathcal{N}^M, s_{i,k}(\eta) = \{O\}} \mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_i^u, \eta), \quad (4.42)$$

and the portion of $\mathbb{F}(\mathbf{x}_i^u)$ that is hidden by the obstacles approximated by the OEM is $\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_i^u) \cap \mathbb{F}(\mathbf{x}_i^u)$.

Figure 4.27 in Section 4.7 illustrates the difference between $\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_i^u) \cap \mathbb{F}(\mathbf{x}_i^u)$ and $\mathbb{H}_{i,k}^{\text{CVS}}$.

4.7 . Simulation

This chapter presents a set-membership estimator that exploits CVS measurements to characterize and update a set estimate containing the location of each target that has been identified. This

estimator is composed of the algorithms that are required to characterize the set estimates $\mathbb{X}_{i,j,k}^{t,m}$, $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$, $\mathbb{X}_{i,k}^o$, and $\mathbb{X}_{i,j,k}^{t,m}$ from the CVS measurements available at time t_k . Sections 4.2 and 4.3 provide an explanation of how the CVS measurements are exploited to get these set estimates, and Appendix A.2 details how they are characterized. Then, these set estimates are used to update the knowledge UAV i has before time t_k regarding the potential locations of the identified and unidentified targets, *i.e.*, $\mathbb{X}_{i,j,k-1}^t$, with $j \in \mathcal{L}_{i,k-1}^t$, and $\overline{\mathbb{X}}_{i,k-1}^t$. These set estimates are also used to update the knowledge UAV i has before time t_k regarding the environment, *i.e.*, $\mathbb{X}_{i,k-1}^o$. This update is explained in Section 4.5.

This section presents some simulation results regarding the performance of the proposed target location estimator. Only one UAV is considered and it only exploits the CVS measurements available at one time instant to estimate the possible locations of the targets and update its previous knowledge. Consequently, to evaluate the performance of the proposed estimator, we consider that $\mathcal{L}_{i,k-1}^t = \emptyset$, and $\overline{\mathbb{X}}_{i,k-1}^t = \mathbb{X}_g$. The tracking of the targets and, thus, the capacity of the UAVs to maintain an accurate estimation of their location is studied in Chapter 5.

Simulation results have been obtained using Webots ([Webots](#),) to generate the targets, the CVS embedded on the UAV, as well as the measurements it collects. Webots then sends the CVS measurements to MATLAB where the code has been implemented.

4.7.1 . Simulation conditions and metrics

The targets are cars with the following dimensions $4.6 \text{ m} \times 1.8 \text{ m} \times 1.5 \text{ m}$. The cylinder \mathbb{C}^t containing the shape of the cars is chosen with a radius $r^t = 2.5 \text{ m}$, and a height $h^t = 1.5 \text{ m}$.

A single UAV is placed within the Rol. It is equipped with a camera producing images that are subsampled to get an image of size $N_r \times N_c = 360 \times 480$ pixels in order to reduce the computational time. The camera has an aperture angle of $\pi/4$ rad, and is fixed in the UAV frame. Therefore, the rotation matrix is

$$\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}_i^c} = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix}, \quad (4.43)$$

where $\theta = \pi/6$ is the orientation of the camera in the UAV body frame.

A depth map is provided by the *range finder* of Webots with the same resolution as the camera. Its maximal measurement range is set to $d_{\max} = 300 \text{ m}$. The bounds of the noise w , appearing in the acquisition of $\mathbf{D}_i(n_r, n_c)$ from $\mathbf{D}_i^o(n_r, n_c)$ in Section 3.5.2, *i.e.*, $\mathbf{D}_i(n_r, n_c) = \mathbf{D}_i^o(n_r, n_c)(1 + w)$ are taken as $[\underline{w}, \overline{w}] = [-0.01, 0.01]$. For example, if $\mathbf{D}_i^o(n_r, n_c) = 200 \text{ m}$, then the depth measurement $\mathbf{D}_i(n_r, n_c)$ is obtained by selecting $w \in [\underline{w}, \overline{w}] = [-0.01, 0.01]$ uniformly at random, to get, for example, $\mathbf{D}_i(n_r, n_c) = 201 \text{ m}$. Then, using (3.35), one gets $[\mathbf{D}_i](n_r, n_c) = [199.01 \text{ m}, 203.03 \text{ m}]$. The pixel classification and target detection-identification is done by Webots.

Figure 4.8 shows an example of CVS information provided to the UAV.

Most of the outputs provided by the target location estimator are set estimates. Consider that a set estimate $\mathbb{X} \subset \mathbb{R}^2$ represents the knowledge a UAV has about the location of a target. The uncertainty regarding its estimated location is measured by the area $\phi(\mathbb{X})$ of \mathbb{X} . A more intuitive evaluation of the

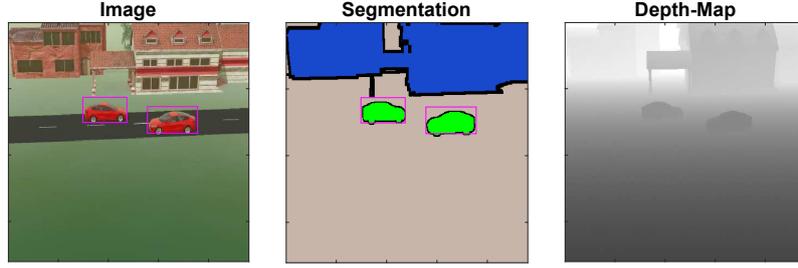


Figure 4.8 – CVS information provided to a UAV by the sensors simulated by Webots. On the left : the image \mathbf{I}_i with two bounding boxes. At the center : the array of pixel labels \mathbf{L}_i with two bounding boxes (*Target*-labeled pixels in green, *Obstacle*-labeled pixels in blue, *Ground*-labeled pixels in brown, and *Unknown*-labeled pixels in black). On the right : the depth map \mathbf{D}_i .

estimation accuracy of a target location may be obtained by the radius

$$\phi^{\text{eR}}(\mathbb{X}) = \sqrt{\frac{\phi(\mathbb{X})}{\pi}}, \quad (4.44)$$

of a disc with an area equal to $\phi(\mathbb{X})$. In addition, if $\mathbf{c}(\mathbb{X})$ represents the barycenter of \mathbb{X} , then the distance between \mathbf{x}_j^{tg} and $\mathbf{c}(\mathbb{X}_{i,j}^{\text{t}})$, *i.e.*,

$$\phi^{\text{c}}(\mathbb{X}_{i,j}^{\text{t}}) = \left\| \mathbf{c}(\mathbb{X}_{i,j}^{\text{t}}) - \mathbf{x}_j^{\text{tg}} \right\|, \quad (4.45)$$

represents an additional metric to evaluate the estimation uncertainty of the location \mathbf{x}_j^{tg} of target j using the set estimate $\mathbb{X}_{i,j}^{\text{t}}$.

The following simulations are performed at a single time instant while assuming no prior knowledge regarding the environment and the targets, *i.e.*, the obstacles and the targets are detected and identified for the first time.

4.7.2 . Location estimation of a fully-visible target

First, we consider the following setup. A UAV is placed at the location $(0, 0, 50)^{\text{T}}$. Its attitude is $(0, 0, 0)^{\text{T}}$. Then, five targets are placed in front of the UAV. The location of these targets are : $(60, 0, 0)^{\text{T}}$, $(90, 0, 0)^{\text{T}}$, $(120, 0, 0)^{\text{T}}$, $(150, 0, 0)^{\text{T}}$, and $(180, 0, 0)^{\text{T}}$. Figure 4.9 illustrates some CVS measurements that UAV i can acquire with this setup. The five targets are identified and the pink rectangle is their associated bounding box $\left[\mathcal{Y}_{i,j}^{\text{t}} \right]$. No obstacle is considered. Additionally, for the sake of simplicity, we consider that the targets are sufficiently far apart to avoid characterizing $\mathbb{X}_{i,j}^{\text{t,m}}$, with $j = 1, \dots, 5$. The usefulness of $\mathbb{X}_{i,j}^{\text{t,m}}$ is considered in Section 4.7.3.

Figure 4.10 illustrates the set estimates obtained considering the setup presented in Figure 4.9. The black dots in each subfigure represent the location of the targets. The blue set represents $\mathbb{P}_i^{\text{g}}(\mathcal{Y}_i^{\text{g}})$, *i.e.*, the part of \mathbb{X}_g containing no target location. It does not contain any black dots. The sets delimited by a red border are the projection on the ground $\mathbf{p}_g(\mathbb{P}_{i,j}^{\text{t}})$ of $\mathbb{P}_{i,j}^{\text{t}}$, *i.e.*, a subset of \mathbb{X}_0 that has a non-empty intersection with the shape of target j , for each target. Since the UAV is able to observe the sunroof of the cars, $\mathbf{p}_g(\mathbb{P}_{i,j}^{\text{t}})$ contains the location of its associated target. Nonetheless, according to Corollary 11,

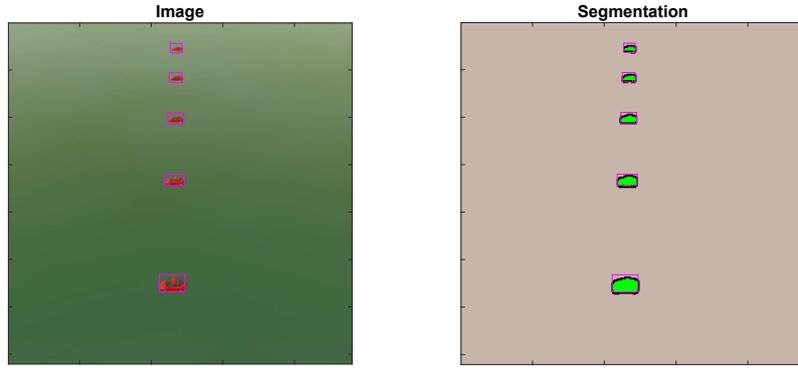


Figure 4.9 – Illustration of some of the CVS measurements that a UAV can acquire in a situation with no obstacles. The UAV location and attitude is $(0, 0, 50, 0, 0, 0)^T$. The location of the targets is : $(60, 0, 0)^T$, $(90, 0, 0)^T$, $(120, 0, 0)^T$, $(150, 0, 0)^T$, and $(180, 0, 0)^T$.

$\mathbb{P}_{i,j}^t$ is only known to intersect the target shape. Therefore, there is no reason that $\mathbf{p}_g(\mathbb{P}_{i,j}^t)$ always contains $\mathbf{x}_j^{t,g}$. The only set that has this property is $\mathbb{X}_{i,j}^{t,m}$ represented by the sets delimited by a green border.

Moreover, as $\mathbb{X}_{i,j}^{t,m}$ is proved to contain $\mathbf{x}_j^{t,g}$ and $\mathbb{P}_i^g(\mathcal{Y}_i^g)$ is proved to not contain $\mathbf{x}_j^{t,g}$, with $j = 1, \dots, 5$, the part $\mathbb{X}_{i,j}^{t,m} \cap \mathbb{P}_i^g(\mathcal{Y}_i^g)$ of $\mathbb{X}_{i,j}^{t,m}$ can be removed because it does not contain $\mathbf{x}_{j,k}^{t,g}$, see the update rule (4.32) in Section 4.5.2.

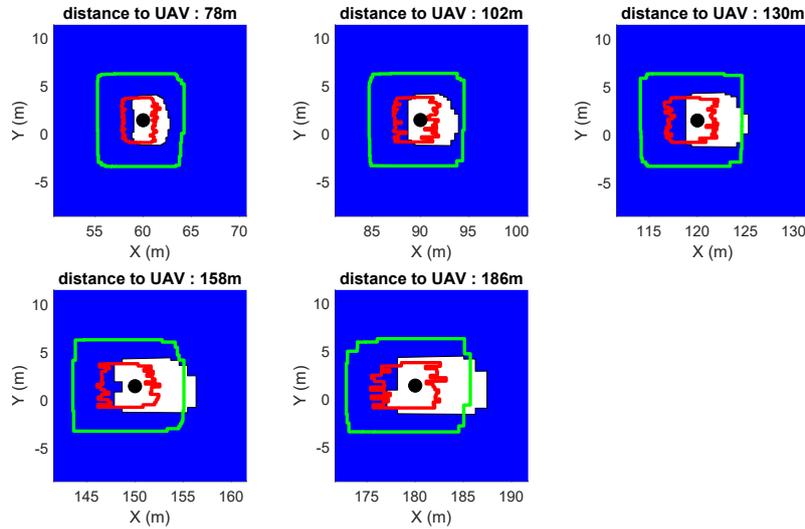


Figure 4.10 – Representation of the set estimates $\mathbb{P}_i^g(\mathcal{Y}_i^g)$ (in blue), $\mathbf{p}_g(\mathbb{P}_{i,j}^t)$ (in red), and $\mathbb{X}_{i,j}^{t,m}$ (in green) obtained after the processing of the CVS measurements with the set-membership estimator. The black dots are the different target locations.

Since it is the first time these five targets are identified, the set estimate containing their possible locations is computed using (4.32). Additionally, there is no obstacles and we assumed that $\bar{\mathbb{X}}_{i,k-1}^t =$

\mathbb{X}_g . Therefore, in this case, (4.32) becomes

$$\mathbb{X}_{i,j}^t = \left(\mathbb{X}_g \cap \mathbb{X}_{i,j}^{t,m} \right) \setminus \mathbb{P}_i^g(\mathcal{Y}_i^g), \quad (4.46)$$

$$= \mathbb{X}_{i,j}^{t,m} \cap \left(\mathbb{X}_g \setminus \mathbb{P}_i^g(\mathcal{Y}_i^g) \right). \quad (4.47)$$

One can remark that $\mathbb{X}_g \setminus \mathbb{P}_i^g(\mathcal{Y}_i^g)$ are the white sets shown in Figure 4.10.

Figure 4.11 compares $\phi^{\text{eR}}(\mathbb{X}_{i,j}^{t,m})$ (in green), and $\phi^{\text{eR}}(\mathbb{X}_{i,j}^t)$ (in black), and $\phi^c(\mathbb{X}_{i,j}^t)$ (in red) obtained for these five targets. The results have been averaged over 100 independent simulations where only the measurement noise associated to the evaluation of the depth map changes. As illustrated in Figures 4.10 and 4.11, the closer the target is to the UAV, the better the accuracy is. This is because, since the UAV has a better point of view for these close targets, a large portion of $\mathbb{X}_{i,j}^{t,m}$ is overlapped by $\mathbb{P}_i^g(\mathcal{Y}_i^g)$.

Regarding $\phi^{\text{eR}}(\mathbb{X}_{i,j}^t)$, one can remark that the standard deviation increases for targets located more than 130 m away from the UAV. As illustrated with the standard deviation of $\phi^{\text{eR}}(\mathbb{X}_{i,j}^{t,m})$, the measurement noise associated to the evaluation of the depth map affects the characterization of $\mathbb{X}_{i,j}^{t,m}$. This is expected since $\mathbb{P}_i((n_r, n_c))$ relies on $[\mathbf{D}_i]$ (n_r, n_c) , with $(n_r, n_c) \in [\mathcal{Y}_{i,j}^t] \cap \mathcal{Y}_i^t$, and is required to get $\mathbb{X}_{i,j}^{t,m}$. Nevertheless, the characterization of $\mathbb{P}_i^g(\mathcal{Y}_i^g)$ does not rely on the depth map. $\mathbb{P}_i^g(\mathcal{Y}_i^g)$ is the same for each of the 100 simulations, but $\mathbb{X}_{i,j}^{t,m}$ varies. Consequently, with (4.46) and Figure 4.10, one can see that for any small variation of the size of $\mathbb{X}_{i,j}^{t,m}$, the set estimate $\mathbb{X}_{i,j}^t$ obtained for the targets that are located at 78 m and 102 m from the UAV will not change. The reason is because $(\mathbb{X}_g \setminus \mathbb{P}_i^g(\mathcal{Y}_i^g)) \cap \mathbb{P}_i^g(\mathcal{Y}_{i,j}^t)$, where $\mathbb{P}_i^g(\mathcal{Y}_{i,j}^t)$ represents the portion of \mathbb{X}_g that is masked by target j , is included in $\mathbb{X}_{i,j}^{t,m}$. In fact, some *Unknown* pixels have to be considered for the characterization of $\mathbb{P}_i^g(\mathcal{Y}_{i,j}^t)$, but no notation is introduced for them. Nevertheless, a small variation of the size of $\mathbb{X}_{i,j}^{t,m}$ for the targets located more than 130 m away from the UAV implies a small variation of $\mathbb{X}_{i,j}^t$ because $(\mathbb{X}_g \setminus \mathbb{P}_i^g(\mathcal{Y}_i^g)) \cap \mathbb{P}_i^g(\mathcal{Y}_{i,j}^t)$ is no longer included in $\mathbb{X}_{i,j}^{t,m}$. The sets $\mathbb{P}_i^g(\mathcal{Y}_{i,j}^t)$ are represented by the white set in each subfigure of Figure 4.10.

Moreover, according to ϕ^c , the error associated with the point estimate $c(\mathbb{X}_{i,j}^t)$ is less than two meters. Additionally, the curve of $\phi^c(\mathbb{X}_{i,j}^t)$ is rising between 78 m and 130 m, *i.e.*, $c(\mathbb{X}_{i,j}^t)$ is getting away from $x_j^{t,g}$, and stagnates afterward. The reason is, between 78 m and 130 m, we have that $(\mathbb{X}_g \setminus \mathbb{P}_i^g(\mathcal{Y}_i^g)) \cap \mathbb{P}_i^g(\mathcal{Y}_{i,j}^t)$ is included in $\mathbb{X}_{i,j}^{t,m}$. Thus $\mathbb{X}_{i,j}^t$ is the portion of the ground that is hidden by the target, *i.e.*, its shadow, and in our case, this shadow of the target is not under but behind the target, see the white sets in Figure 4.10. Moreover, the further the target is from the UAV, the wider the shadow is. Consequently, $c(\mathbb{X}_{i,j}^t)$ is getting away from $x_j^{t,g}$. Nevertheless, between 130 m and 186 m, we no longer have $(\mathbb{X}_g \setminus \mathbb{P}_i^g(\mathcal{Y}_i^g)) \cap \mathbb{P}_i^g(\mathcal{Y}_{i,j}^t)$ included in $\mathbb{X}_{i,j}^{t,m}$. Thus, $\mathbb{X}_{i,j}^t$ no longer grows behind the target with its shadow along the x axis. Consequently, $\phi^c(\mathbb{X}_{i,j}^t)$ stagnates.

In Figure 4.12, the set estimates $\mathbb{X}_{i,j}^t$ are represented in yellow and the point estimates $c(\mathbb{X}_{i,j}^t)$ are represented by a red dot. The circle of center $c(\mathbb{X}_{i,j}^t)$ and radius $\phi^{\text{eR}}(\mathbb{X}_{i,j}^t)$ (in red) provides a

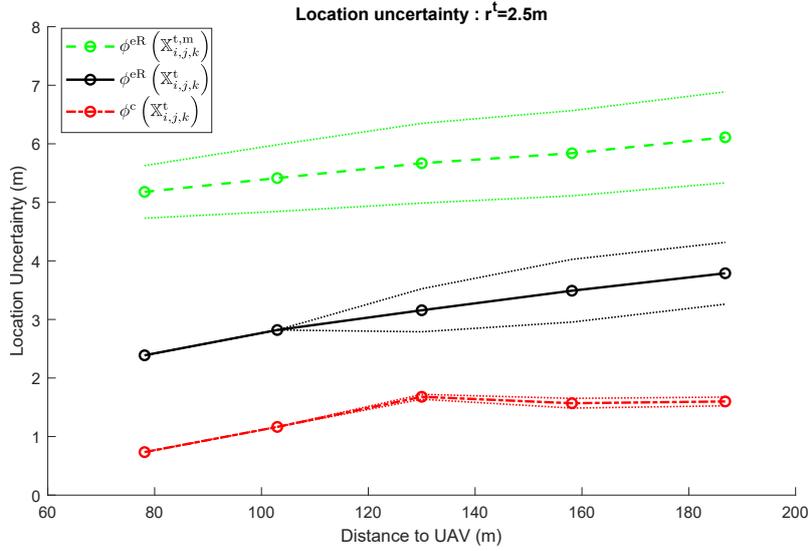


Figure 4.11 – Mean value and standard deviation over 100 simulations of $\phi^{eR}(\mathbb{X}_{i,j}^{t,m})$ (in green), $\phi^{eR}(\mathbb{X}_{i,j}^t)$ (in black), and $\phi^c(\mathbb{X}_{i,j}^t)$ (in red).

good approximation of $\mathbb{X}_{i,j}^t$ and of the associated estimation uncertainty.

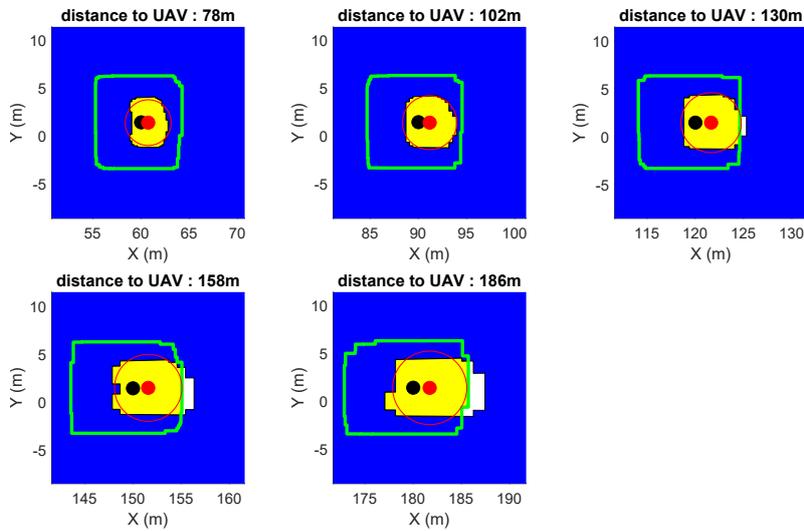


Figure 4.12 – Representation of the set estimates $\mathbb{P}_i^g(\mathcal{Y}_i^g)$ (in blue), $\mathbb{X}_{i,j}^{t,m}$ (in green), and $\mathbb{X}_{i,j}^t$ (in yellow). The black dots represent the different target locations. The red dots correspond to $c(\mathbb{X}_{i,j}^t)$ and the circle of center $c(\mathbb{X}_{i,j}^t)$ and radius $\phi^{eR}(\mathbb{X}_{i,j}^t)$ is in red.

Figure 4.13 compares the performance of the estimator for different values of the radius r^t of the cylinder \mathbb{C}^t . Set apart the vertical translation, the evolution of the $\phi^{eR}(\mathbb{X}_{i,j}^{t,m})$ (the green curves) are almost the same. A green curve only differs from another one due to a translation that is seemingly

equal to the difference of r^t between the two green curves. Consequently, the larger r^t is, *i.e.*, the larger the uncertainty about the target shape is, the larger the size of the set $\mathbb{X}_{i,j}^{t,m}$ will be. Nonetheless, the value of r^t only impacts the size of $\mathbb{X}_{i,j}^{t,m}$ but not the size of $\mathbb{P}_i^g(\mathcal{Y}_i^g)$. Therefore, as shown by the solid curves (black curves), most of the additional uncertainty of the estimated target location is compensated (the black curves are almost identical). This compensation of the additional uncertainty related to a large value of r^t comes from the capacity of eliminating the portions of $\mathbb{X}_{i,j}^{t,m}$ that cannot contain the location of the target, *i.e.*, $\mathbb{X}_{i,j}^{t,m} \cap \mathbb{P}_{i,k}^g(\mathcal{Y}_i^g)$.

Regarding $\phi^{eR}(\mathbb{X}_{i,j}^t)$ and $\phi^c(\mathbb{X}_{i,j}^t)$, the larger r^t becomes, the more probable $\mathbb{X}_{i,j}^{t,m}$ contains the set $(\mathbb{X}_g \setminus \mathbb{P}_i^g(\mathcal{Y}_i^g)) \cap \mathbb{P}_i^g(\mathcal{Y}_{i,j}^t)$, as illustrated with $\phi^{eR}(\mathbb{X}_{i,j}^{t,m})$. This is the reason why the standard deviation of $\phi^{eR}(\mathbb{X}_{i,j}^t)$ increases later, and why the distance from the UAV at which $\phi^c(\mathbb{X}_{i,j}^t)$ stagnates is greater.

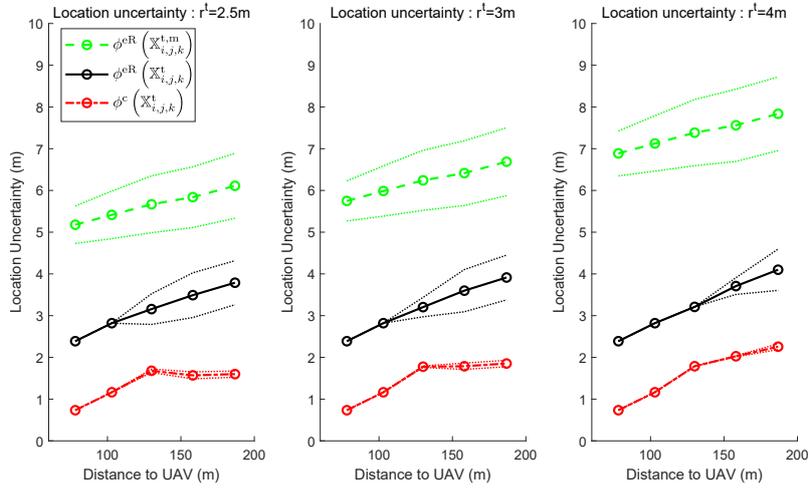


Figure 4.13 – Mean value with standard deviation over 100 simulations of $\phi^{eR}(\mathbb{X}_{i,j}^{t,m})$ (in green), $\phi^{eR}(\mathbb{X}_{i,j}^t)$ (in black), and $\phi^c(\mathbb{X}_{i,j}^t)$ (in red). The left subfigure is for $r^t = 2.5$ m. The middle subfigure is for $r^t = 3$ m. The right subfigure is for $r^t = 4$ m.

The impact on the measurement noise associated to the depth map is now analyzed. We consider three bounded sets $[\underline{w}, \bar{w}]$ for that noise : $[\underline{w}, \bar{w}] = [-0.01, 0.01]$, $[\underline{w}, \bar{w}] = [-0.05, 0.05]$, and $[\underline{w}, \bar{w}] = [-0.1, 0.1]$. Figure 4.14 compares the different metrics that are obtained for each of these bounded sets. As expected, the uncertainty regarding $\mathbb{X}_{i,j}^{t,m}$, *i.e.*, $\phi^{eR}(\mathbb{X}_{i,j}^{t,m})$, increases when we considered a larger bounded set for $[\underline{w}, \bar{w}]$. Actually, the larger $[\underline{w}, \bar{w}]$ is, the larger $\mathbb{P}_i((n_r, n_c))$ is for any $(n_r, n_c) \in [\mathcal{Y}_{i,j}^t] \cap \mathcal{Y}_i^t$. Thus, $\mathbb{P}_{i,j}^t$ is larger which leads to have a larger $\mathbb{X}_{i,j}^{t,m}$.

Nonetheless, and as in Figure 4.13, the value of $[\underline{w}, \bar{w}]$ only impacts the size of $\mathbb{X}_{i,j}^{t,m}$ but not the size of $\mathbb{P}_i^g(\mathcal{Y}_i^g)$. Therefore, most of the additional uncertainty of the estimated target location is compensated by $\mathbb{P}_i^g(\mathcal{Y}_i^g)$. Consequently, $\phi^{eR}(\mathbb{X}_{i,j}^t)$ and $\phi^c(\mathbb{X}_{i,j}^t)$ are almost unaffected by the measurement noise associated to the depth map.

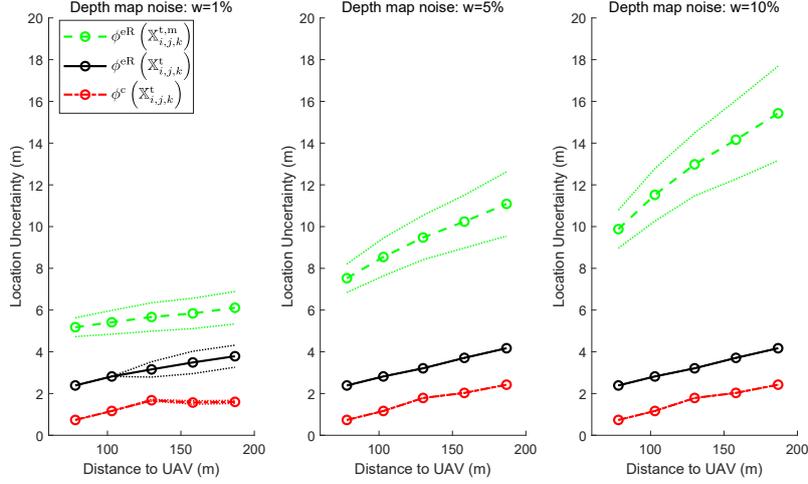


Figure 4.14 – Mean value with standard deviation over 100 simulations of $\phi^{\text{eR}}(\mathbb{X}_{i,j}^{\text{t},m})$ (in green), $\phi^{\text{eR}}(\mathbb{X}_{i,j}^{\text{t}})$ (in black), and $\phi^{\text{c}}(\mathbb{X}_{i,j}^{\text{t}})$ (in red). The left subfigure is for $[\underline{w}, \bar{w}] = [-0.01, 0.01]$. The middle subfigure is for $[\underline{w}, \bar{w}] = [-0.05, 0.05]$. The right subfigure is for $[\underline{w}, \bar{w}] = [-0.1, 0.1]$.

The impact on the estimation uncertainty of the camera resolution is now analyzed, considering a camera acquiring images of $N_r \times N_c = 360 \times 480$ pixels and a camera with a resolution of $N_r \times N_c = 720 \times 960$ pixels. Figure 4.15 displays the obtained CVS measurements with a camera of resolution 720×960 and Figure 4.16 shows the set estimates obtained with that camera. Figure 4.17 compares the different metrics obtained with the camera of $N_r \times N_c = 360 \times 480$ pixels (in blue) and that of $N_r \times N_c = 720 \times 960$ pixels (in green).

Comparing $\phi^{\text{eR}}(\mathbb{X}_{i,j}^{\text{t},m})$ (dashed curves) for the two cameras, one observes that a larger resolution decreases slightly the estimation uncertainty. This is also true for $\phi^{\text{eR}}(\mathbb{X}_{i,j}^{\text{t}})$. Besides, since there are more pixels labeled *Ground* or *Target*, the characterization of $\mathbb{P}_i^{\text{g}}(\mathcal{Y}_i^{\text{g}})$ is more accurate to the ground truth and $\mathbb{X}_{i,j}^{\text{t},m}$ is likely to be larger. Consequently, it is more likely to have $(\mathbb{X}_{\text{g}} \setminus \mathbb{P}_i^{\text{g}}(\mathcal{Y}_i^{\text{g}})) \cap \mathbb{P}_i^{\text{g}}(\mathcal{Y}_{i,j}^{\text{t}})$ included in $\mathbb{X}_{i,j}^{\text{t},m}$. That explains the difference between the two curves associated to $\phi^{\text{c}}(\mathbb{X}_{i,j}^{\text{t}})$.

4.7.3 . Exploiting the minimum distance between targets

A problem with target localization is that, when a target is identified, the UAV gets no information of what is behind the identified target. Therefore, without Assumption (3.11), the update of $\bar{\mathbb{X}}_{i,k-1}^{\text{t}}$, according to (4.29), is

$$\bar{\mathbb{X}}_{i,k}^{\text{t}} = \mathbb{X}_{\text{g}} \setminus \mathbb{P}_{i,k}^{\text{g}}(\mathcal{Y}_{i,k}^{\text{g}}). \quad (4.48)$$

Consequently, during the update of $\bar{\mathbb{X}}_{i,k-1}^{\text{t}}$, the portion of $\bar{\mathbb{X}}_{i,k|k-1}^{\text{t}}$ (which is equal to \mathbb{X}_{g} in (4.48)) that is hidden by a target, *i.e.*, $\bar{\mathbb{X}}_{i,k|k-1}^{\text{t}} \cap \mathbb{P}_i^{\text{g}}(\mathcal{Y}_{i,j}^{\text{t}})$, cannot be removed using $\mathbb{P}_{i,k}^{\text{g}}(\mathcal{Y}_{i,k}^{\text{g}})$. Figure 4.18 illustrates this phenomenon where $\bar{\mathbb{X}}_{i,k}^{\text{t}} \cap \mathbb{P}_i^{\text{g}}(\mathcal{Y}_{i,j}^{\text{t}})$ are represented by yellow sets.

To address this problem, we introduce in Section 3.2.1 an assumption regarding the minimum

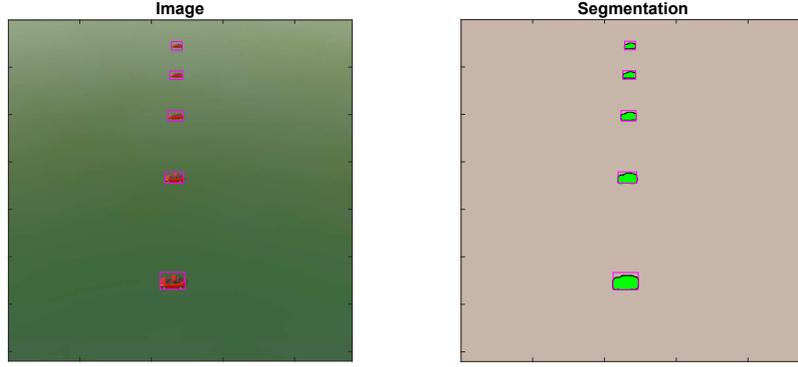


Figure 4.15 – Illustration of CVS measurements with a camera of resolution $N_r \times N_c = 720 \times 960$ pixels

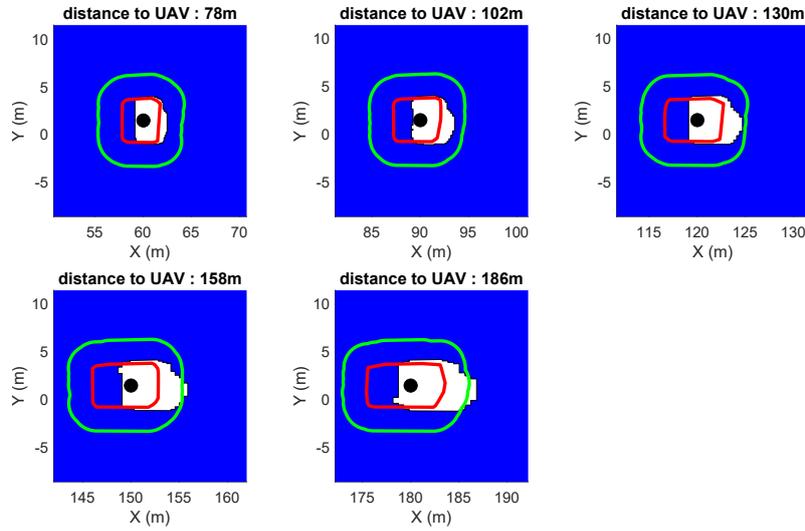


Figure 4.16 – Representation of the set estimates $\mathbb{P}_i^g(\mathcal{Y}_i^g)$ (in blue), $\mathbf{p}_g(\mathbb{P}_{i,j}^t)$ (in red), and $\mathbb{X}_{i,j}^{t,m}$ (in green) that are obtained from the measurements provided by the embedded CVS when the camera provides images of resolution $N_r \times N_c = 720 \times 960$ pixels. The black dots are the different target locations.

safety distance between two targets. This hypothesis has been exploited in Proposition 18 to define a set $\mathbb{X}_{i,j}^{t,m}$ that, once a target j has been identified, does not contain any other target locations except the location of that identified target. This section evaluates the impact of this assumption.

Consider $r_5^{tt} = 6$ m. This distance represents the minimum distance separating the location of two vehicles, *i.e.*, $\mathbf{x}_{j,k}^{t,g}$ and $\mathbf{x}_{\ell,k}^{t,g}$ with $j \in \mathcal{N}^t$ and $\ell \in \mathcal{N}^t \setminus \{j\}$. Therefore, by accounting for the dimensions of the considered targets, *i.e.*, $4.6 \text{ m} \times 1.8 \text{ m}$, it is as if the vehicles have to maintain at least a distance between 1.4 m (if they are face to face) and 4.2 m (if they are side by side) between their chassis. Figure 4.18 shows $\mathbb{P}_{i,k}^g(\mathcal{Y}_i^g)$ (in blue), $\mathbb{X}_{i,k}^t \cap \mathbb{P}_{i,k}^g(\mathcal{Y}_{i,j,k}^t)$ (in yellow), and the set estimate $\mathbb{X}_{i,j,k}^{t,m}$ (in green), that does not contain any target location other than $\mathbf{x}_{j,k}^{t,g}$. The set $\mathbb{X}_{i,j,k}^{t,m}$ (in green) can be used to remove a portion of the set $\mathbb{X}_{i,k}^t$ (in yellow), as illustrated in Figure 4.18.

To evaluate the percentage of size reduction of the set $\mathbb{X}_{i,k}^t \cap \mathbb{P}_{i,k}^g(\mathcal{Y}_{i,j,k}^t)$ using $\mathbb{X}_{i,j,k}^{t,m}$, consider the

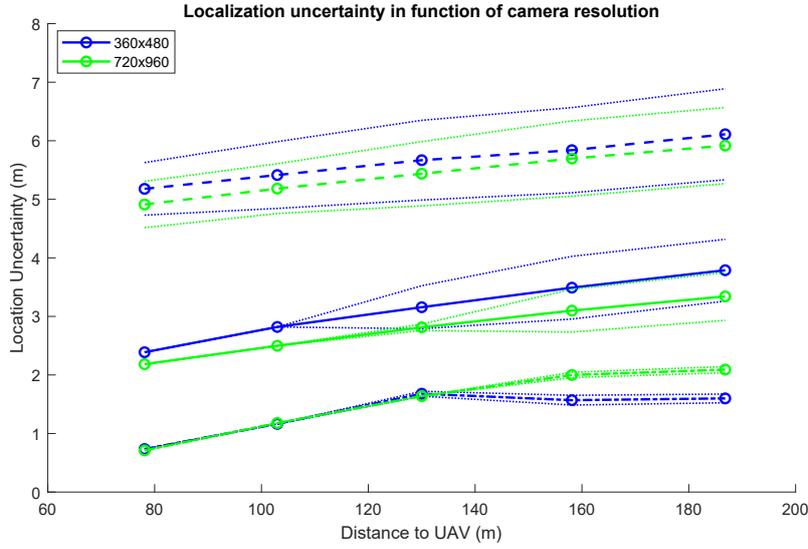


Figure 4.17 – Mean value and standard deviation over 100 simulations of $\phi^{eR}(\mathbb{X}_{i,j}^{t,m})$ (dashed lines - curves at the top of the figure), $\phi^{eR}(\mathbb{X}_{i,j}^t)$ (solid lines - curves in the middle of the figure), and $\phi^c(\mathbb{X}_{i,j}^t)$ (broken lines - curves at the bottom of the figure). The green curves represent the results obtained with a camera of resolution $N_r \times N_c = 720 \times 960$ pixels; the blue curves represent the results obtained with a camera of resolution $N_r \times N_c = 360 \times 480$ pixels.

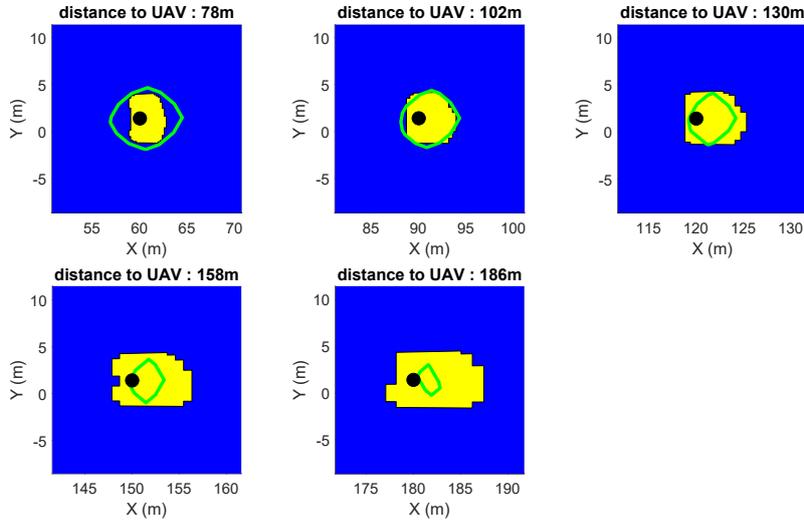


Figure 4.18 – Representation of the set estimates $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$ (in blue), $\bar{\mathbb{X}}_{i,k}^t \cap \mathbb{P}_{i,k}^g(\mathcal{Y}_{i,j,k}^t)$ (in yellow), and $\mathbb{X}_{i,j,k}^t$ (in green). The black dots are the different target locations. Here, $r_s^{tt} = 6$ m

metric

$$\Phi^\Delta(\mathbb{X}_1, \mathbb{X}_2) = 100\phi(\mathbb{X}_1 \cap \mathbb{X}_2) / \phi(\mathbb{X}_1) \quad (4.49)$$

to evaluate how much \mathbb{X}_2 overlaps \mathbb{X}_1 .

Figure 4.19 shows $\Phi^\Delta \left(\mathbb{X}_{i,j,k}^{t,m}, \bar{\mathbb{X}}_{i,k}^t \cap \mathbb{P}_{i,k}^g \left(\mathcal{Y}_{i,j,k}^t \right) \right)$ for different values of r_s^{tt} . When $r_s^{tt} = 8$ m, the set $\mathbb{X}_{i,j,k}^{t,m}$ entirely covers $\bar{\mathbb{X}}_{i,k}^t \cap \mathbb{P}_{i,k}^g \left(\mathcal{Y}_{i,j,k}^t \right)$ for targets less than 130 m away from the UAV. Thus, according to Proposition 18, $\bar{\mathbb{X}}_{i,k}^t \cap \mathbb{P}_{i,k}^g \left(\mathcal{Y}_{i,j,k}^t \right)$ is proved to contain no target. Regarding the other values of r_s^{tt} that have been considered for the simulation, one can conclude that Assumption 3.11 can be used to prove that most of, if not all, the portion of $\bar{\mathbb{X}}_{i,k}^t$ that is hidden behind a target, *i.e.*, $\bar{\mathbb{X}}_{i,k}^t \cap \mathbb{P}_{i,k}^g \left(\mathcal{Y}_{i,j,k}^t \right)$, can be safely removed.

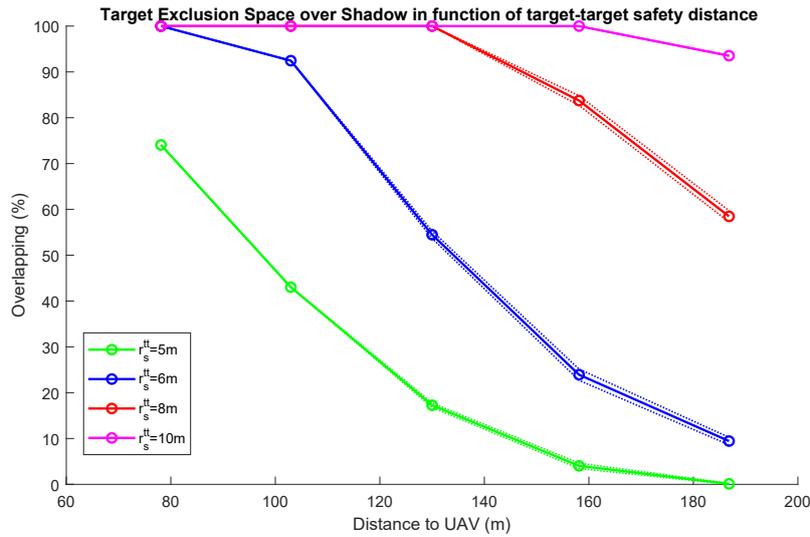


Figure 4.19 - Mean value with standard deviation over 100 simulations of $\Phi^\Delta \left(\mathbb{X}_{i,j,k}^{t,m}, \bar{\mathbb{X}}_{i,k}^t \cap \mathbb{P}_{i,k}^g \left(\mathcal{Y}_{i,j,k}^t \right) \right)$ for different values of r_s^{tt} .

4.7.4 . Target localization in the presence of obstacles

This section presents some results for the localization of a target in the presence of obstacles.

Figure 4.20 shows the identification of a target that appears close to an obstacle in the image acquired by the camera of the UAV. This is clear in the bottom-left and bottom-right subfigures, where the pixels labeled *Target* (in green) are close to those labeled *Obstacle* (in blue).

Figure 4.21 shows that because of an obstacle, only a small portion of \mathbb{X}_g can be proved to be target-free using Proposition 14 (the white set in Figure 4.21). To avoid overloading the figure, $\mathbb{X}_{i,j,k}^{t,m}$ is not represented here.

In Figure 4.20, the target is fully visible : no part of it is hidden behind an obstacle. We assume that it is the first time this target is identified. Consequently, the set $\mathbb{X}_{i,j,k}^t$ containing all of its possible locations must be characterized using (4.32). Nonetheless, as the target appears to be close to an obstacle, $\mathbb{X}_{i,j,k}^{t,m}$ (the set with a pink border) overlaps with the portion of $\bar{\mathbb{X}}_{i,k|k-1}^t \setminus \mathbb{P}_{i,k}^g \left(\mathcal{Y}_{i,k}^g \right)$ (the yellow set) that is hidden behind the obstacle. Consequently, even though the target is fully visible, some of its possible locations contained in $\mathbb{X}_{i,j,k}^t$ are located behind the obstacle.

4.7.5 . Inner approximation of the r_s^{to} -ground neighborhood of an obstacle

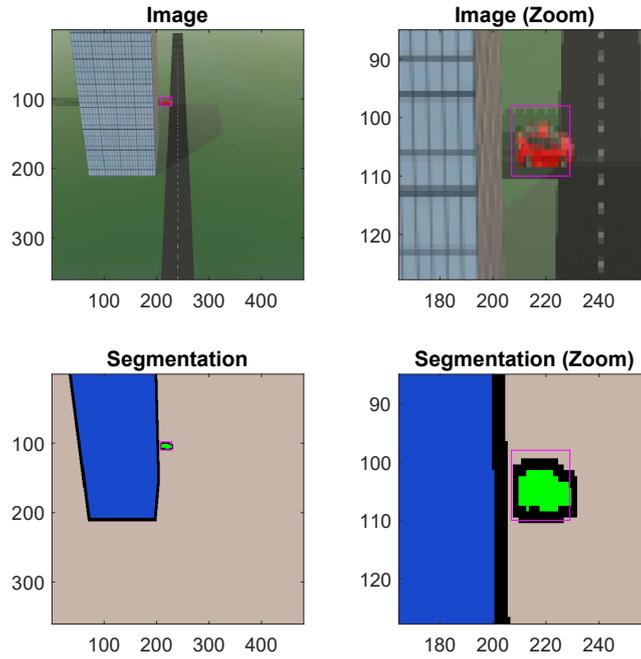


Figure 4.20 – Illustration of the CVS measurements when a target is located close to an obstacle (in the image).

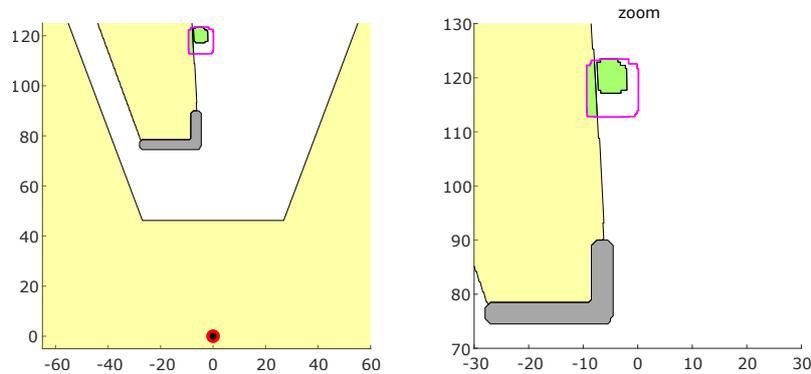


Figure 4.21 – Estimation of the location of a target which seems to be close to an obstacle (c.f. Figure 4.20). The red dot represents the UAV location. $\bar{\mathbb{X}}_{i,k}^t$ is in yellow, $\mathbb{X}_{i,j,k}^t$ is in green, and $\mathbb{X}_{i,j,k}^{t,m}$ is represented by the set with the pink border. The set $\bar{\mathbb{X}}_{i,k}^o$ representing an inner approximation of r_s^{to} -ground neighborhood of an obstacle is in gray.

In Section 3.2.1, assumption (3.8) states that the minimum distance between the projection on the ground of the shape of an obstacle and the location of a target should be at least r_s^{to} . This distance is interpreted as a safety distance between a target and an obstacle to avoid collisions. Using (3.8), Proposition 17 has introduced the set $\bar{\mathbb{X}}_i^o$ which is the inner-approximation of the r_s^{to} -ground neighborhood of all obstacles in the environment. In Proposition 17, $\bar{\mathbb{X}}_i^o$ has been shown to target-free. Thus, as indicated by the equations in Section 4.5.2, this set can be used by a UAV to gain new knowledge about the environment in order to improve the accuracy of the target location estimator. This section

shows the capacity of a UAV located at $(0, 0, 50)$ to characterize $\underline{\mathbb{X}}_i^o$ from CVS measurements reported in Figure 4.22.

The considered environment contains towers and houses, as well as a long synthetic building in light blue used as a test bench. Different values for r_s^{to} are considered for the characterization of $\underline{\mathbb{X}}_i^o$, see Figure 4.23.

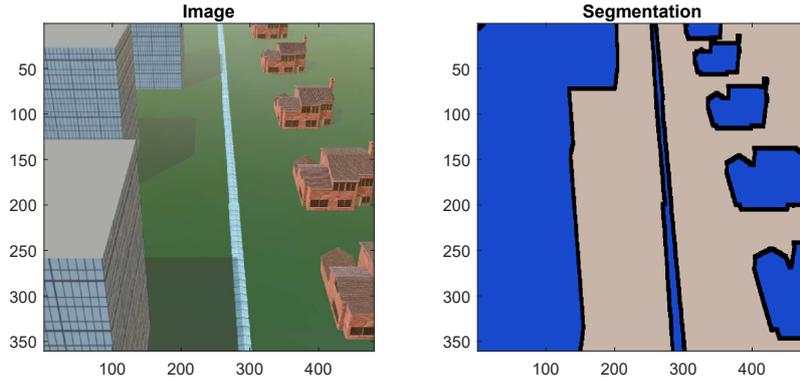


Figure 4.22 – Illustration of the CVS measurements when numerous obstacles are located within the FoV of the UAV.

Figure 4.23 shows the sets $\underline{\mathbb{X}}_i^o$ that are obtained using Proposition 17 and the pixels labeled *Obstacle* illustrated in Figure 4.22 for four values of r_s^{to} : 2 m (in green), 2.5 m (in red), 3 m (in blue), and 5 m (in purple).

For each r_s^{to} , the UAV is able to characterize $\underline{\mathbb{X}}_i^o$. One can remark that, when r_s^{to} is large, the UAV is unable to characterize an inner approximation of the r_s^{to} -ground neighborhood of the obstacles that are far away from it. This is illustrated Figure 4.23 (right) where the green and red sets start to disappear. Nonetheless, since the obstacles are statics, the UAV does not need to characterize an inner-approximation of the r_s^{to} -ground neighborhood of far-away obstacles at each time step. During target search, as illustrated in Figure 5.13 of Chapter 5, the UAVs will explore the RoI in search for a target. Consequently, they will, at some point, get closer to the obstacles that were far away at a previous time instant, and be able to characterize an inner approximation of the r_s^{to} -ground neighborhood of those obstacles.

4.7.6 . OEM construction

As detailed in Section 4.6, the CVS measurements are also exploited to update a 2.5D map in order to get an approximation of the location, height, and shape of the obstacles located in the RoI. To show how the OEM is updated from the CVS measurements available at a given time instant, we consider a situation where a UAV is currently observing some buildings and houses. Figure 4.24 shows the environment that is considered for this simulation. Figure 4.25 illustrates the CVS measurements that are available to the UAV.

Figure 4.26 illustrates the OEM $\mathcal{M}_{i,k} = \{c(\eta), s_{i,k}(\eta), h_{i,k}(\eta)\}_{\eta \in \mathcal{N}^M}$ while assuming that at time t_{k-1} we had $s_{i,k-1}(\eta) = \{U\}$, i.e., the cell η is *Unexplored*, and $h_{i,k-1}(\eta) = 0$ for all cells $\eta \in \mathcal{N}^M$. The resolution of the OEM is 5 m. The CVS measurements related to the detected obstacles and the CVS measurements related to the ground are exploited by the UAV to find and update some of the

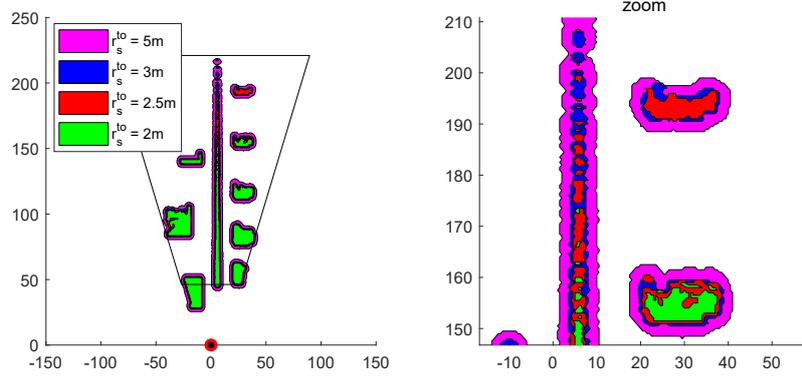


Figure 4.23 – Illustration of \mathbb{X}_i^0 obtained for different values of $r_s^{t_0}$. The red dot represents the UAV location and the set with the black border is $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$.



Figure 4.24 – Example of a scene simulated on Webots where several buildings and houses are located within the RoI

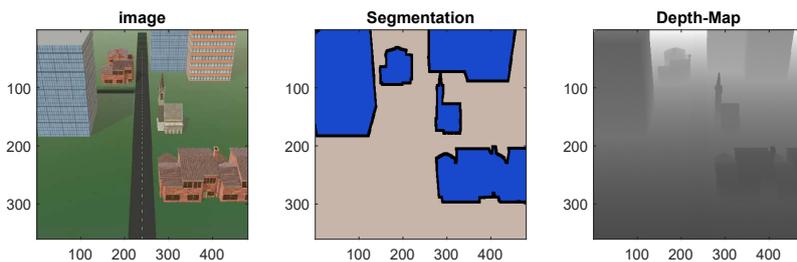


Figure 4.25 – CVS measurements available to the UAV when it observes the scene in Figure 4.24. The UAV state (location and attitude) is $(0, 0, 50, 0, 0, 0)^T$.

cells of $\mathcal{M}_{i,k}$ using (4.39) and (4.40). The cells whose status is updated from *Unexplored* to *Empty* are green in Figure 4.26. Similarly, the cells whose status is updated from *Unexplored* to *Occupied* are red. The height of each parallelepiped is the maximum height measured for an obstacle that could

be associated with one of the *occupied* cells. Finally, blue squares represent the cells that are still *Unexplored*.

The UAV is able to observe the roof of the blue building on its left, see Figure 4.25. Consequently, it is able to get a better estimation of its height and shape than with the two obstacles in the background for which the UAV does only see one face without seeing their roof. The blue building on the left-hand side is represented in Figure 4.26 by the block composed of 16 parallelepipeds around the XY -coordinates (100, 50). The buildings in the background are represented by the series of 10 parallelepipeds that are around the XY -coordinates (150, -50).

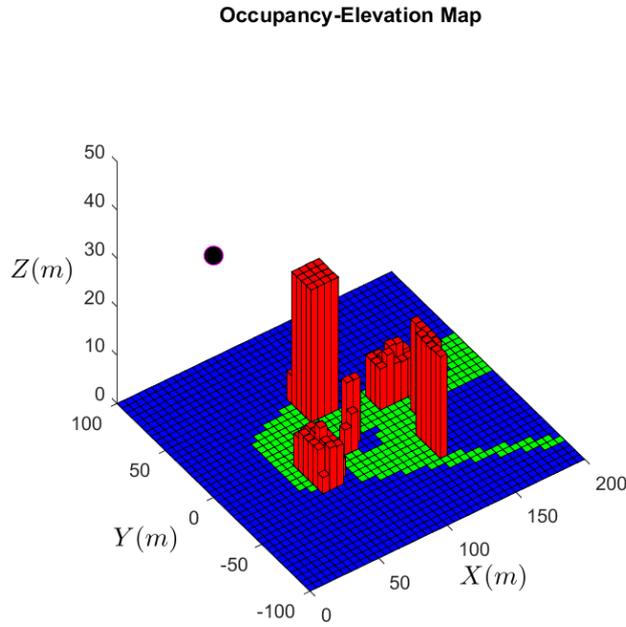


Figure 4.26 – Representation of an OEM after processing the CVS measurements reported in Figure 4.25. The blue squares represent the *Unexplored* cells. The green squares represent the *Empty* cells. The red parallelepipeds result from the *Occupied* cells and the registered height. The UAV location is represented by the black dot with a pink outline.

With the OEM reported in Figure 4.26, one can compare $\mathbb{H}_i^{\text{CVS}}$, *i.e.*, the approximation of the portion of the ground hidden by the obstacles, targets, or unknown objects, with $\mathbb{H}_i^{\text{OEM}}(\mathbf{x}_i^u) \cap \mathbb{F}(\mathbf{x}_i^u)$, *i.e.*, the portion of the ground hidden by the obstacles approximated by the OEM. To evaluate the similarity between $\mathbb{H}_i^{\text{CVS}}$ and $\mathbb{H}_i^{\text{OEM}}(\mathbf{x}_i^u) \cap \mathbb{F}(\mathbf{x}_i^u)$, we consider the metric

$$\text{IoU}(\mathbb{A}, \mathbb{B}) = \frac{\phi(\mathbb{A} \cap \mathbb{B})}{\phi(\mathbb{A} \cup \mathbb{B})}, \quad (4.50)$$

that computes the ratio between the area of the intersection between two sets and the area of their union. The better \mathbb{B} approximates \mathbb{A} , the closer $\text{IoU}(\mathbb{A}, \mathbb{B})$ is to 1.

Figure 4.27 (right) illustrates $\mathbb{H}_i^{\text{CVS}}$ (in blue) and $\mathbb{H}_i^{\text{OEM}}(\mathbf{x}_i^u) \cap \mathbb{F}(\mathbf{x}_i^u)$ (in red). Figure 4.27 (left) shows the black boxes used to approximate the shape and the location of the obstacles that are currently

under observation, and the $\mathbb{H}_i^{\text{OEM}}(\mathbf{x}_i^u) \cap \mathbb{F}(\mathbf{x}_i^u)$ that is obtained with these boxes. In Figure 4.27 (right), $\mathbb{H}_i^{\text{CVS}}$ overlays $\mathbb{H}_i^{\text{OEM}}(\mathbf{x}_i^u) \cap \mathbb{F}(\mathbf{x}_i^u)$. By evaluating IoU ($\mathbb{H}_i^{\text{CVS}}, \mathbb{H}_i^{\text{OEM}}(\mathbf{x}_i^u) \cap \mathbb{F}(\mathbf{x}_i^u)$), we obtain a similarity between $\mathbb{H}_i^{\text{CVS}}$ and $\mathbb{H}_i^{\text{OEM}}(\mathbf{x}_i^u) \cap \mathbb{F}(\mathbf{x}_i^u)$ of 91.7 %. Consequently, from this one-time measurement, an OEM of resolution 5m can be used to reliably approximate $\mathbb{H}_i^{\text{CVS}}$.

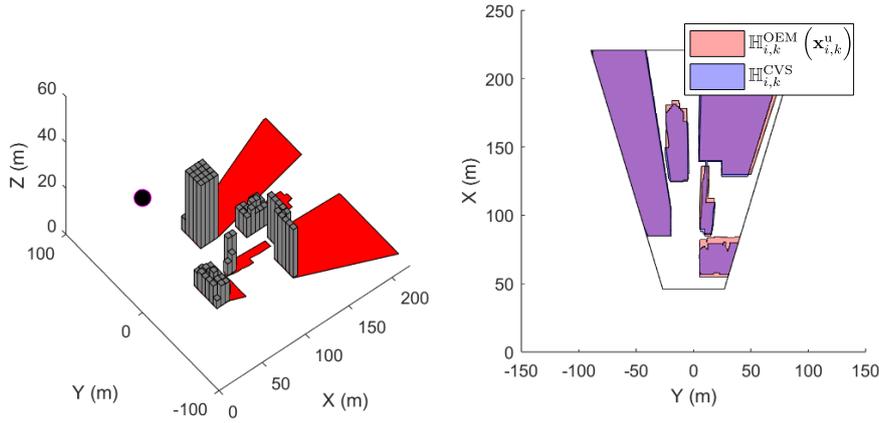


Figure 4.27 – Illustration of $\mathbb{H}_i^{\text{OEM}}(\mathbf{x}_i^u) \cap \mathbb{F}(\mathbf{x}_i^u)$ (in red) and $\mathbb{H}_i^{\text{CVS}}$ (in blue) when the OEM is built with only the CVS measurements acquired at time t_k . In the left subfigure, the black boxes represent the set of boxes (4.41), and the red set is $\mathbb{H}_i^{\text{OEM}}(\mathbf{x}_i^u)$, where \mathbf{x}_i^u is the black dot. In the right subfigure, $\mathbb{H}_i^{\text{CVS}}$ overlaid $\mathbb{H}_i^{\text{OEM}}(\mathbf{x}_i^u) \cap \mathbb{F}(\mathbf{x}_i^u)$.

4.8 . Conclusion

This chapter presents the first (to the best of our knowledge) set-membership target location estimator exploiting multiple measurements provided by a CVS. Compared to previous works, the CVS in (Jaulin, 2009; Ehambram et al., 2022) consists of a sonar or a LiDAR. In (Kenmogne et al., 2019), the CVS returns an image, and the pinhole model is used to model the camera, but no depth map or labeled image is considered.

The main difficulty with the CVS is that it is difficult to obtain measurement models because the CVS often involves deep learning algorithms. To address this problem, we introduced in Chapter 3 several assumptions to relate the CVS measurements (labeled pixels, depth map, bounding boxes around identified targets) to the environment, the obstacles, and the targets. Under these assumptions, we propose several approaches to exploit the CVS measurements in order to solve a target localization problem.

In this chapter, Section 4.2 presents how the measurements related to an identified target j can be used to characterize a set estimate $\mathbb{X}_{i,j,k}^{t,m}$ that is guaranteed to contain the location of that target.

This is only true if the hypotheses on the model associated with the range measurements, the pixel labeling, the target identification, and the *a priori* knowledge the UAV has regarding the dimensions of the target, *i.e.*, (3.4), are satisfied. The simulation results in Figure 4.13 and Figure 4.14 show that the larger the uncertainty on the target dimensions or on the depth-map noise, the more inaccurate the set estimate $\mathbb{X}_{i,j,k}^{t,m}$. To reduce the target location uncertainty of $\mathbb{X}_{i,j,k}^{t,m}$, a UAV can exploit CVS measurements related to the ground or the obstacles as negative information in order to characterize two set estimates, *i.e.*, $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$ and $\mathbb{X}_{i,k}^o$, that are guaranteed to contain no target location, see Section 4.3. Then, Section 4.5.2 details how $\mathbb{X}_{i,j,k}^{t,m}$ can be combined with $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$ and $\mathbb{X}_{i,k}^o$ to get a set estimate $\mathbb{X}_{i,j,k}^t$. This set estimate is guaranteed to contain the true location of target j and, as illustrated in Figure 4.14, the use of $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$ contributes to reduce the localization uncertainty. With assumption (3.11) in Chapter 3 a UAV can also characterize another set estimate that is guaranteed to contain no target location except one, but this assumption relies on *a priori* known minimum distance between two targets that may not be always verified in practice. Overall, the performance of the target location estimator, according to the above simulations, is satisfactory. The point estimation error, *i.e.*, $\phi^c(\mathbb{X}_{i,j,k}^t)$, for a one-time observation, is about ± 1.5 meter with a point estimate uncertainty, *i.e.*, $\phi^{eR}(\mathbb{X}_{i,j,k}^t)$ that looks linear with the distance between the target and the UAV.

Section 4.4 presents how the CVS measurements that are not related to the ground can be exploited to approximate the portion of the ground that is masked by a target, an obstacle, or something else. The set estimate $\mathbb{H}_{i,k}^{CVS}$ that approximates this hidden ground can be used to approximate the portion of the ground that is masked by the obstacles at time t_k . Nonetheless, it is hard to use this set estimate to get some knowledge about the shape, the height, or the location of the obstacles. The set estimate $\mathbb{X}_{i,k}^o$, which approximates a neighborhood around the obstacles, can be used to get an approximation of the location of the detected obstacles, but their locations only. To get some knowledge about the shape of the obstacles, we present in Section 4.6 an approach that relies on an OEM. The OEM exploits the CVS measurements related to detected obstacles to update either the *unexplored* or the *occupied* cells in order to represent locally the detected obstacles using parallelepipeds, see Figure 4.26. The CVS measurements related to the ground are also exploited to characterize a region without obstacles.

Chapter 5 will present a control strategy which allows to solve the CSAT problem in an unknown environment. In this next chapter, each UAV embeds the target location estimator presented in Chapter 4 to determine, from their location at time t_k who and where are the targets that are identified and where the unidentified targets may be. Then, after the update of the set estimates $\mathbb{X}_{i,j,k}^t$, $\overline{\mathbb{X}}_{i,k}^t$, and $\mathbb{X}_{i,k}^o$, using the prediction-correction approach detailed in Section 4.5.2, the UAVs cooperatively update their trajectories to keep searching for new targets while maintaining an accurate estimation of the location of identified targets. We propose to use the OEM to improve the trajectory update step in this next chapter. Chapter 6 presents an adaptation of the estimator to account for an uncertain UAV state.

Chapter 5 - Trajectory design for target search and tracking

This chapter presents several strategies for iteratively designing the UAV trajectories, so that the fleet can efficiently explore the Region of Interest (RoI) to search and track targets. As the RoI is cluttered with obstacles, the UAVs should fly toward areas where targets are more likely to be found. The main difficulty is that the shapes and locations of those obstacles are *a priori* unknown and they might occlude the Field of View (FoV) of the UAVs.

Each UAV runs a target location estimator to exploit CVS measurements to characterize and update set estimates of the location of identified targets and sets that do not contain any target location, see Chapter 4. This chapter shows how these sets are used to update the trajectory of each UAV. As will be seen, the target search and tracking efficiency depends on the knowledge each UAV is able to gain about its environment.

A distributed approach is considered to improve the robustness of the fleet against the loss of one or several UAVs. The design could be made on each UAV independently of the other UAVs. Nevertheless, to improve the exploration and target tracking efficiency of the fleet, each UAV may take into account the updated trajectories of its neighbors to design its own trajectory. When the targets outnumber the UAVs, a trade-off needs to be found between exploration to find new targets, and recapture and pursuit of previously identified targets to maintain accurate estimates of their locations.

The trajectory update that solves this trade-off is performed by an adaptation of a distributed Model Predictive Control (MPC) approach introduced in (Ibenthal et al., 2023). Each UAV searches for a sequence of control inputs that minimizes a criterion reflecting the goals to achieve. Using a model of the UAV dynamics (3.13), the sequence of control inputs can be transformed into a sequence of states representing a trajectory. The trajectory such a the considered criterion serves as a reference and it is exploited by a low-level controller implemented in each UAV.

The estimator presented in Chapter 4 is used to estimate and update the possible locations of the targets. Nevertheless, accounting for all sources of information provided by the CVS may be complex. Consequently, we first do not consider all hypotheses introduced in Chapter 3. Thus, some of the set estimates presented in Chapter 4 cannot be characterized, which also affects the trajectory design. Then, more hypotheses are considered and more complex trajectory design strategies are introduced. For example, one ignores that the targets must evolve within the RoI while keeping a minimal safety distance with the obstacles, see Hypothesis (3.8) in Chapter 3. Consequently, the characterization of an area free of targets around the detected obstacles is not possible, see Section 4.3.2 of Chapter 4.

Section 5.1 formulates the trajectory design problem. Then, Section 5.2 presents four different strategies to update the trajectories of the UAVs. Section 5.3 addresses some practical issues related to the implementation of the MPC approach. Finally, Section 5.4 evaluates the performance of the presented strategies in a simulated urban-like environment.

5.1 . Problem formulation

At time t_k , UAV i uses the depth map $\mathbf{D}_{i,k}$, the pixels labeled *Target* $\mathcal{Y}_{i,k}^t$, and a bounding box $[\mathcal{Y}_{i,j,k}^t]$ associated to an identified target j to characterize, with Proposition 13, a set estimate $\mathbb{X}_{i,j,k}^{t,m}$ that is guaranteed to contain the location of that target. Then, UAV i exploits the pixels labeled *Ground* $\mathcal{Y}_{i,k}^g$ and the Hypothesis (3.54) to characterize with Proposition 14 a set estimate $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$ that contains no target location. Using pixels labeled *Obstacles* $\mathcal{Y}_{i,k}^o$, the depth map $\mathbf{D}_{i,k}$, and Hypothesis (3.8) (minimum distance between targets and obstacles), UAV i is also able to characterize with Proposition 17 a region $\mathbb{X}_{i,k}^o$ around the detected obstacles that will never contain any target location. Additionally, from $\mathbb{X}_{i,j,k}^{t,m}$ and using Hypothesis (3.11) (minimum distance between targets), UAV i can characterize with Proposition 18 another set estimate $\underline{\mathbb{X}}_{i,j,k}^{t,m}$ that cannot contain the location of any target except the location of target j .

Once these set estimates have been characterized, UAV i can update its knowledge regarding the environment and the possible location of the identified and unidentified targets. This update is performed via a prediction-correction process that also exploits the information broadcast by the neighbors of UAV i within communication range. The update is detailed in Section 4.5 of Chapter 4. At the end of the prediction-correction process, UAV i has access, at time t_k , to a set estimate $\mathbb{X}_{i,j,k}^t$ containing the possible locations of an identified target $j \in \mathcal{L}_{i,k}^t$. These set estimates are gathered in the list $\mathcal{X}_{i,k}^t = \left\{ \mathbb{X}_{i,j,k}^t \right\}_{j \in \mathcal{L}_{i,k}^t}$. It has also access to a set estimate $\overline{\mathbb{X}}_{i,k}^t$ containing all possible locations of targets still to identify as well as a set $\mathbb{X}_{i,k}^o$ representing the portion of the RoI that can never contain a target location due to the presence of an obstacle.

From pixels that are not labeled *Ground*, UAV i can also characterize a set estimate $\mathbb{H}_{i,k}^{\text{CVS}}$. This set represents an approximation of the part of \mathbb{X}_g that UAV i has been unable to observe at time t_k due to the presence of obstacles, targets, or unidentified elements in the acquired image, see Section 4.4 of Chapter 4. Finally, UAV i can exploit pixels labeled *Ground* or *Obstacle* to build and update an OEM $\mathcal{M}_{i,k}$ that provides an approximate representation of the detected obstacles, see Section 4.6 of Chapter 4. From the OEM, UAV i can estimate the part of the environment hidden by obstacles $\mathbb{H}_{i,k}^{\text{OEM}}$, see Section 4.6.4.

Using part or all of this available knowledge, UAV i has to update its trajectory to explore the RoI and find new targets while maintaining an accurate estimation of the location of already identified targets. This update may account for the information broadcast by its neighbors $\mathcal{N}_{i,k}$ to design a trajectory complementary to that of UAVs in $\mathcal{N}_{i,k}$.

A criterion is introduced to evaluate the efficiency of the designed trajectories. As in (Ibenthal et al., 2023), the criterion is based on the area of the set estimates $\mathbb{X}_{i,j,k}^t$, with $j \in \mathcal{L}_{i,k}^t$, and $\overline{\mathbb{X}}_{i,k}^t$. The area of the set $\mathbb{X}_{i,j,k}^t$ provides a measure of the localization uncertainty for target $j \in \mathcal{L}_{i,k}^t$. The area of $\overline{\mathbb{X}}_{i,k}^t$ determines the estimation uncertainty of the location of targets still to be identified.

Consequently, at time t_k , the target location estimation uncertainty for UAV i is expressed as

$$\Phi \left(\mathcal{X}_{i,k}^t, \overline{\mathbb{X}}_{i,k}^t \right) = \phi \left(\overline{\mathbb{X}}_{i,k}^t \cup \bigcup_{j \in \mathcal{L}_{i,k}^t} \mathbb{X}_{i,j,k}^t \right), \quad (5.1)$$

where $\phi(\mathbb{X})$ is the area of the set $\mathbb{X} \subset \mathbb{R}^2$. The target estimation uncertainty for the fleet of UAVs is then the average of the target location estimation uncertainty for each UAV

$$\bar{\Phi}_k = \frac{1}{N^u} \sum_{i \in \mathcal{N}^u} \Phi(\mathcal{X}_{i,k}^t, \bar{\mathbb{X}}_{i,k}^t). \quad (5.2)$$

The minimization of (5.2) leads to a trade-off between tracking already identified targets to maintain an accurate estimation of their location (by reducing the size of $\mathbb{X}_{i,j,k}^t, j \in \mathcal{L}_{i,k}^t$), and exploring the Rol to search for unidentified targets (to reduce the size of $\bar{\mathbb{X}}_{i,k}^t$). The transition between search and tracking modes depends on the relative sizes of the sets $\mathbb{X}_{i,j,k}^t, j \in \mathcal{L}_{i,k}^t$ and $\bar{\mathbb{X}}_{i,k}^t$.

Depending on the computing power of the UAVs, it is possible that some of the aforementioned set estimates cannot be characterized. This limitation affects simultaneously the target location estimator and the trajectory update. In Section 5.2 three situations are considered. In Section 5.2.1, we assume that the UAVs do not have the time to process the pixels labeled *Obstacles* and, thus, $\mathbb{X}_{i,k}^o$ as well as $\mathbb{H}_{i,k}^{CVS}$ are not evaluated. In Section 5.2.2, we assume that the UAVs have the computing power to characterize all the set estimates but are unable to build and update the OEM. This limitation is not considered in Sections 5.2.3 (and 5.2.4). Thus, the most complex target location estimator and the most complex trajectory design can be used to search and track the targets. Table 5.1 summarizes which set estimates are assumed available during each of the proposed trajectory update strategies.

Sections	Set estimates provided by the estimator					Evaluation of the OEM $\mathcal{M}_{i,k}$
	$\mathcal{X}_{i,k}^t / \bar{\mathbb{X}}_{i,k}^t$	$\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$	$\underline{\mathbb{X}}_{i,j,k}^{t,m}$	$\mathbb{X}_{i,k}^o$	$\mathbb{H}_{i,k}^{CVS}$	
5.2.1	X	X	X	-	-	-
5.2.2	X	X	X	X	X	-
5.2.3 and 5.2.4	X	X	X	X	X	X

Table 5.1 – Set estimates provided by the set-membership estimator presented in Chapter 4 depending on the limitations that are considered in some of the sections of Chapter 5.

5.2 . Trajectory design for cooperative exploration

This section presents several variants of a cooperative trajectory design algorithm for the fleet of UAVs. Each variant depends on the type of processed information, which impacts the required computing resources for each UAV.

5.2.1 . Trajectory design ignoring the presence of obstacles

In this section, it is assumed that the UAVs are unable to maintain a map of the Rol. Moreover, Hypothesis (3.8) related to the minimum distance between a target and an obstacle is not exploited. Consequently, $\mathbb{X}_{i,k}^o$ is not evaluated and $\mathbb{X}_{i,k}^o = \emptyset$.

With such a setup, the trajectory design strategy consists in driving the UAVs toward areas which may contain targets, *i.e.*, either $\bar{\mathbb{X}}_{i,k}^t$ or $\mathbb{X}_{\ell,j,k}^t$ with $j \in \mathcal{L}_{\ell,k}^t$, so as to minimize (5.2). This strategy is simple but may be suboptimal in the presence of obstacles. Figure 5.1 illustrates such a situation

where, starting from the red UAV location, three candidate future locations (in blue) are considered. Without knowledge of the map, they would theoretically lead to the same reduction of the size of $\bar{\mathbb{X}}_{i,k}^t$ (in yellow). In practice, the trajectory leading to the left yields a significantly reduced FoV due to the gray obstacle.

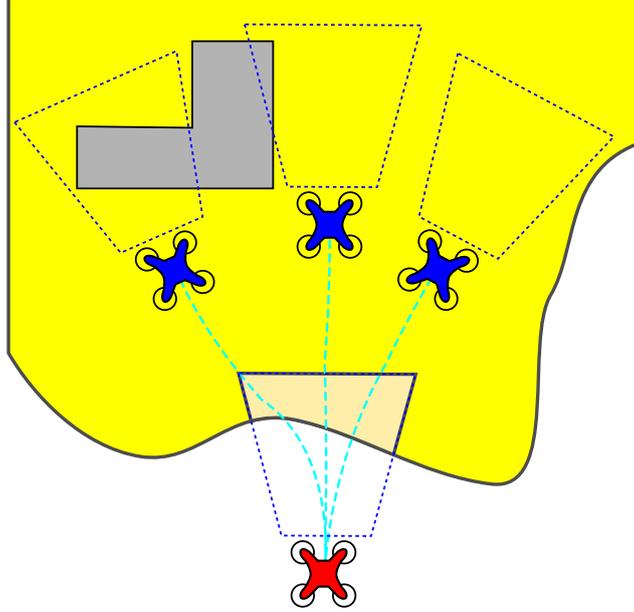


Figure 5.1 – Illustration of the design of a trajectory to explore the set $\bar{\mathbb{X}}_{i,k}^t$ (in yellow) when the UAV (in red) is not aware of the presence of an obstacle (in gray). The blue UAVs represent three possible next UAV locations.

Evaluation of the trajectory

At time t_{k-1} , assume that UAV i has evaluated a trajectory over a prediction horizon of $h \geq 1$ time intervals $\hat{\mathbf{x}}_{i,k-1:h|k-1}^u = (\hat{\mathbf{x}}_{i,k|k-1}^u, \dots, \hat{\mathbf{x}}_{i,k-1+h|k-1}^u)$. With the CVS measurements obtained at time t_k , UAV i updates its set estimates and its trajectory to explore the RoI more efficiently, by taking the information acquired at time t_k into account. This update is done by using a distributed MPC, see (Christofides et al., 2013), implemented on UAV i . The MPC approach determines a sequence of h control inputs $\hat{\mathbf{u}}_{i,k:h} = (\hat{\mathbf{u}}_{i,k}, \dots, \hat{\mathbf{u}}_{i,k+h-1}) \in \mathbb{U}^h$ that minimizes the predicted target localization estimation uncertainty denoted $\bar{\Phi}_{k+h|k}$ at time t_{k+h} , i.e.,

$$\hat{\mathbf{u}}_{i,k:h} = \arg \min_{\mathbf{u}_{i,k:h}} (\bar{\Phi}_{k+h|k}) \quad (5.3)$$

with

$$\bar{\Phi}_{k+h|k} = \frac{1}{N^u} \sum_{\ell \in \mathcal{N}^u} \Phi \left(\mathcal{X}_{\ell,k+h|k}^{t,P}, \bar{\mathbb{X}}_{\ell,k+h|k}^{t,P} \right), \quad (5.4)$$

and

$$\Phi \left(\mathcal{X}_{i,k+h|k}^{t,P}, \overline{\mathbb{X}}_{i,k+h|k}^{t,P} \right) = \phi \left(\overline{\mathbb{X}}_{i,k+h|k}^{t,P} \cup \bigcup_{j \in \mathcal{L}_{i,k+h|k}^{t,P}} \mathbb{X}_{i,j,k+h|k}^{t,P} \right). \quad (5.5)$$

In (5.5), $\mathcal{L}_{i,k+h|k}^{t,P}$ is the list of target indexes that UAV i predicts to be identified at time t_{k+h} knowing $\mathcal{L}_{i,k}^t$. The set estimate $\mathbb{X}_{i,j,k+h|k}^{t,P}$ contains all the predicted locations of the target $j \in \mathcal{L}_{i,k+h|k}^{t,P}$ at time t_{k+h} knowing $\mathbb{X}_{i,j,k}^t$. The set $\overline{\mathbb{X}}_{i,k+h|k}^{t,P}$ contains all the predicted locations of the targets still to identify at time t_{k+h} knowing $\overline{\mathbb{X}}_{i,k}^t$. The list $\mathcal{X}_{i,k+h|k}^{t,P}$ gathers all the $\mathbb{X}_{i,j,k+h|k}^{t,P}$. The evaluation of (5.5) is achieved while accounting for the trajectories broadcast by the other UAVs. Once $\hat{\mathbf{u}}_{i,k:h}$ is available, an updated trajectory $\hat{\mathbf{x}}_{i,k:h|k}^u = \left(\hat{\mathbf{x}}_{i,k+1|k}^u, \dots, \hat{\mathbf{x}}_{i,k+h|k}^u \right)$ for UAV i can be deduced using the dynamic model of the UAV (3.13). The elements in the sequence of predicted states $\hat{\mathbf{x}}_{i,k:h|k}^u$ are then used as desired state vectors to a low-level controller of UAV i , responsible of tracking the desired trajectory $\hat{\mathbf{x}}_{i,k:h|k}^u$.

UAV i , however, does not have access to all the terms in (5.4). At time t_k , UAV i has only access to its own knowledge and the information broadcast by its neighbors $\mathcal{N}_{i,k}$. Thus, for all UAV ℓ with $\ell \in \mathcal{N}^u \setminus \mathcal{N}_{i,k}$, i.e., the UAVs that are not within the communication range of UAV i , UAV i does not know neither their actual location nor their updated trajectory. Consequently, UAV i cannot predict with which UAVs it will be able to communicate in the future and what knowledge they will share. Moreover, other limitations related to the evaluation of (5.5) must be considered. For instance, UAV i is unable to predict precisely in which state it will observe again target j , and thus it is unable to predict $\mathbb{X}_{i,j,k+h}^{t,m}$ introduced in (4.4). It is also unable to predict $\mathbb{P}_{i,k+h}^g \left(\mathcal{Y}_{i,k+h}^g \right)$, see (4.6). To address these issues, we use the same approach as in (Ibenthal et al., 2021).

Consequently, at time t_k , UAV i updates its trajectory by evaluating a sequence of control inputs $\hat{\mathbf{u}}_{i,k:h} \in \mathbb{U}^h$ that accounts only for its own information by minimizing

$$J_i^0(\mathbf{u}_{k:h}) = \Phi \left(\mathcal{X}_{i,k+h|k}^{t,P}, \overline{\mathbb{X}}_{i,k+h|k}^{t,P} \right), \quad (5.6)$$

where Φ has been introduced in (5.1). For a candidate sequence of control inputs $\mathbf{u}_{k:h} \in \mathbb{U}^h$, the evaluation of $J_i^0(\mathbf{u}_{k:h})$ requires an evaluation of $\mathcal{L}_{i,k+h|k}^{t,P}$, $\overline{\mathbb{X}}_{i,k+h|k}^{t,P}$ and $\mathbb{X}_{i,j,k+h|k}^{t,P}$ with $j \in \mathcal{L}_{i,k+h|k}^{t,P}$ using the information available at time t_k . This is detailed in the next section.

Prediction of estimates using UAV own information

This section describes the evaluation of $\mathbb{X}_{i,j,k+h|k}^{t,P}$, for $j \in \mathcal{L}_{i,k+h|k}^{t,P}$ and $\overline{\mathbb{X}}_{i,k+h|k}^{t,P}$ with a complexity manageable in a MPC context.

First, with the known UAV dynamics introduced by (3.13), and for a sequence of control inputs $\mathbf{u}_{k:h} = (\mathbf{u}_{i,k}, \dots, \mathbf{u}_{i,k+h-1})$, one gets for any $\tau = 1, \dots, h$

$$\mathbf{x}_{i,k+\tau|k}^u = \mathbf{f}^u \left(\mathbf{x}_{i,k+\tau-1|k}^u, \mathbf{u}_{k+\tau-1} \right), \quad (5.7)$$

with $\mathbf{x}_{i,k|k}^u = \mathbf{x}_{i,k}^u$. Once $\mathbf{x}_{i,k+\tau|k}^u$ is available, the resulting FoV $\mathbb{F} \left(\mathbf{x}_{i,k+\tau|k}^u \right)$ can be evaluated using (3.21) and (3.24).

Then, to evaluate $\mathbb{X}_{i,j,k+\tau|k}^{t,P}$, for $j \in \mathcal{L}_{i,k}^t$, and $\overline{\mathbb{X}}_{i,k+\tau|k}^{t,P}$, for $\tau = 1, \dots, h$, we do not account for any potential observation of target j . Thus, as in (Ibenthal et al., 2021), we assume that $\mathcal{D}_{i,k+\tau}^t = \emptyset$, and, according to (4.33), that $\mathcal{L}_{i,k+\tau|k}^{t,P} = \mathcal{L}_{i,k}^t$ for $\tau = 1, \dots, h$. Moreover, we assume that the FoV during the evolution of UAV i will not be limited by any obstacle, target, or unknown object, *i.e.*,

$$\mathbb{P}_{i,k+\tau|k}^g \left(\mathcal{Y}_{i,k+\tau|k}^g \right) = \mathbb{F} \left(\mathbf{x}_{i,k+\tau|k}^u \right) \cap \mathbb{X}_g. \quad (5.8)$$

To evaluate $\mathbb{X}_{i,j,k+\tau|k}^{t,P}$ for $\tau = 1, \dots, h$, from $\mathbb{X}_{i,j,k}^t$ we use an adaptation of the prediction-correction process presented in Section 4.5 of Chapter 4. The prediction (4.24) is unchanged, but, since we assume $\mathcal{D}_{i,k+\tau}^t = \emptyset$, $\mathbb{X}_{i,k}^o = \emptyset$, and with the simplification (5.8), the correction (4.30) can be adapted to evaluate iteratively $\mathbb{X}_{i,j,k+\tau|k}^{t,P}$, for $\tau = 1, \dots, h$, as

$$\mathbb{X}_{i,j,k+\tau|k}^{t,P} = \left(\mathbf{f}^t \left(\mathbb{X}_{i,j,k+\tau-1|k}^{t,P}, [\mathbf{v}^t] \right) \cap \mathbb{X}_g \right) \setminus \left(\mathbb{F} \left(\mathbf{x}_{i,k+\tau|k}^u \right) \cap \mathbb{X}_g \right). \quad (5.9)$$

with $\mathbb{X}_{i,j,k|k}^{t,P} = \mathbb{X}_{i,j,k}^t$.

The same hypotheses are used to evaluate iteratively $\overline{\mathbb{X}}_{i,k+\tau|k}^{t,P}$, $\tau = 1, \dots, h$, from $\overline{\mathbb{X}}_{i,k}^t$ as follows

$$\overline{\mathbb{X}}_{i,k+\tau|k}^{t,P} = \left(\mathbf{f}^t \left(\overline{\mathbb{X}}_{i,k+\tau-1|k}^{t,P}, [\mathbf{v}^t] \right) \cap \mathbb{X}_g \right) \setminus \left(\mathbb{F} \left(\mathbf{x}_{i,k+\tau|k}^u \right) \cap \mathbb{X}_g \right), \quad (5.10)$$

with $\overline{\mathbb{X}}_{i,k|k}^{t,P} = \overline{\mathbb{X}}_{i,k}^t$.

Remark 19. The evaluation of $\mathbb{X}_{i,j,k+\tau|k}^{t,P}$ and $\overline{\mathbb{X}}_{i,k+\tau|k}^{t,P}$ using (5.9) and (5.10) are similar. Moreover, the cost function (5.5) involves the union of these sets predicted at time t_{k+h} . Consequently, introducing

$$\mathbb{X}_{i,k}^A = \overline{\mathbb{X}}_{i,k}^t \cup \bigcup_{j \in \mathcal{L}_{i,k}^t} \mathbb{X}_{i,j,k}^t, \quad (5.11)$$

(5.9) or (5.10) can be directly applied to $\mathbb{X}_{i,k}^A$ to perform an iterative evaluation of

$$\mathbb{X}_{i,k+\tau|k}^{A,P} = \overline{\mathbb{X}}_{i,k+\tau|k}^{t,P} \cup \bigcup_{j \in \mathcal{L}_{i,k}^t} \mathbb{X}_{i,j,k+\tau|k}^{t,P} \quad (5.12)$$

using

$$\mathbb{X}_{i,k+\tau|k}^{A,P} = \left(\mathbf{f}^t \left(\mathbb{X}_{i,k+\tau-1|k}^{A,P}, [\mathbf{v}^t] \right) \cap \mathbb{X}_g \right) \setminus \left(\mathbb{F} \left(\mathbf{x}_{i,k+\tau|k}^u \right) \cap \mathbb{X}_g \right), \quad (5.13)$$

with $\mathbb{X}_{i,k|k}^{A,P} = \mathbb{X}_{i,k}^A$.

Using information from neighbors

Assume that the neighbors of UAV i with indexes in the set $\underline{\mathcal{N}}_{i,k} \subset \mathcal{N}_{i,k} \setminus \{i\}$ have already evaluated their sequence of control inputs $\hat{\mathbf{u}}_{\ell,k:h}$, with $\ell \in \underline{\mathcal{N}}_{i,k}$ and broadcast their sequence of states $\hat{\mathbf{x}}_{\ell,k:h|k}^u$.

After communication, UAV i , during its own control design, may account for the areas $\mathbb{F}(\mathbf{x}_{\ell, k+\tau|k}^u) \cap \mathbb{X}_g$ that will be explored by these neighboring UAVs.

Using the information provided by its neighbors in $\mathcal{N}_{i,k}$, UAV i can anticipate the part of the environment that will be observed by the other UAVs and evaluate $\mathbb{X}_{i, k+\tau|k}^{A,P}$ for $\tau = 1, \dots, h$, as

$$\begin{aligned} \mathbb{X}_{i, k+\tau|k}^{A,P} = & \left(\mathbf{f}^t \left(\mathbb{X}_{i, k+\tau-1|k}^{A,P}, [\mathbf{v}^t] \right) \cap \mathbb{X}_g \right) \\ & \setminus \left(\left(\mathbb{F}(\mathbf{x}_{i, k+\tau|k}^u) \cup \bigcup_{\ell \in \mathcal{N}_{i,k}} \mathbb{F}(\hat{\mathbf{x}}_{\ell, k+\tau|k}^u) \right) \cap \mathbb{X}_g \right), \end{aligned} \quad (5.14)$$

instead of (5.9) and (5.10).

More details related to the organization of communications and computations are provided in Section 5.3.2.

First criterion

In some cases, it is not possible to determine a trajectory which minimizes (5.6). For instance, once large parts of the environment have been explored, UAV i may be located too far from other set estimates to predict that they will be explored in the future h time steps. To address this problem, the following term can be added to (5.6)

$$J_i^{1a}(\mathbf{u}_{k:h}) = d \left(c \left(\mathbb{F}(\mathbf{x}_{i, k+h|k}^{u,P}) \cap \mathbb{X}_g \right), \mathbb{X}_{i, k+h|k}^{A,P} \right) \quad (5.15)$$

where $c(\mathbb{X})$ is the barycenter of the set \mathbb{X} . The idea is to drive the FoV of UAV i toward the closest predicted set estimate.

Combining (5.6) and (5.15), the first considered strategy aims at minimizing the criterion

$$J_i^{\text{STR1}}(\mathbf{u}_{k:h}) = J_i^0(\mathbf{u}_{k:h}) + \lambda J_i^{1a}(\mathbf{u}_{k:h}). \quad (5.16)$$

The parameter λ trades off the reduction of the size of the set estimates via $J_i^0(\mathbf{u}_{k:h})$ and future reductions via $J_i^{1a}(\mathbf{u}_{k:h})$. This parameter can be taken as constant, as in (Ibenthal et al., 2021), or variable as in (Li et al., 2024). For instance, until $J_i^0(\mathbf{u}_{k:h})$ does not remain constant regardless of the tested sequence, $J_i^{1a}(\mathbf{u}_{k:h})$ can be disabled (and thus it is not computed). Otherwise, $J_i^{1a}(\mathbf{u}_{k:h})$ is enabled in order to fly toward the closest set estimate.

This first strategy is an adaptation of the MPC proposed in (Ibenthal et al., 2021) to address a CSAT problem with an unknown cluttered environment. The target location estimator in (Ibenthal et al., 2021) does not consider CVS measurements related to the obstacles. The MPC approach in (Ibenthal et al., 2021) does not exploit $\mathbb{X}_{i,k}^0$, i.e., the portion of the ground that cannot contain any target location due to the presence of an obstacle, see Table 5.1. In the following sections, we consider that pixels labeled *Obstacle* can be exploited by the UAVs. We propose new strategies, built upon that of (Ibenthal et al., 2021), to search for trajectories that account for part of the environment that will be occluded when UAVs move. In (Ibenthal et al., 2023), the obstacles are not explicitly taken into account and the proposed approach relies on controlling the UAVs to fly in a specific formation in order to explore the RoI without gaining environmental knowledge. The UAVs will not be forced to stay in a specific formation in the following sections.

5.2.2 . Considering obstacles

In this section, the UAVs have the computing power to process the pixels labeled *Obstacles* at each time step and, thus, characterize $\mathbb{X}_{i,k}^o$, with $i \in \mathcal{N}^u$, i.e., a region that cannot contain any target location due to the presence of an obstacle. Moreover, using CVS measurements at time t_k , UAV i has also the time to process other CVS measurements to characterize $\mathbb{H}_{i,k}^{\text{CVS}}$, i.e., the part of the ground that UAV i is unable to observe due to the presence of an obstacle, a target, or an unidentified object, see Section 4.4. The OEM is again not taken into account in this section.

As in Section 5.2.1, the detection and identification of targets is ignored. Moreover, as UAV i cannot determine whether new obstacles will be detected in its FoV, the prediction $\mathbb{X}_{i,k+\tau|k}^o$ of $\mathbb{X}_{i,k+\tau}^o$ will be taken equal to $\mathbb{X}_{i,k}^o$ for $\tau = 1, \dots, h$. Similarly, the prediction $\mathbb{H}_{i,k+\tau|k}^{\text{CVS}}$ of $\mathbb{H}_{i,k+\tau}^{\text{CVS}}$ is also assumed constant and equal to $\mathbb{H}_{i,k}^{\text{CVS}}$. This second simplification is coarser than the first, as $\mathbb{H}_{i,k+\tau}^{\text{CVS}}$ depends more on the location of UAV i than $\mathbb{X}_{i,k+\tau}^o$.

Therefore, by using the simplification $\mathbb{H}_{i,k+\tau|k}^{\text{CVS}} = \mathbb{H}_{i,k}^{\text{CVS}}$, (5.8) becomes

$$\mathbb{P}_{i,k+\tau|k}^g \left(\mathcal{Y}_{i,k+\tau|k}^g \right) = \left(\mathbb{F} \left(\mathbf{x}_{i,k+\tau|k}^u \right) \cap \mathbb{X}_g \right) \setminus \mathbb{H}_{i,k}^{\text{CVS}}. \quad (5.17)$$

Then, as for (5.9), the prediction (4.24) is used to predict the evolution of the targets and is corrected by an adaption of (4.30). This adaption accounts for the simplifications we made to evaluate $\mathbb{X}_{i,k+\tau|k}^{\text{A,P}}$ with a complexity manageable in a MPC context, i.e., $\mathcal{D}_{i,k+\tau}^t = \emptyset$, $\mathbb{X}_{i,k+\tau|k}^o = \mathbb{X}_{i,k}^o$, and (5.17). Consequently, with those simplifications, UAV i can adapt (5.13) to evaluate iteratively $\mathbb{X}_{i,k+\tau|k}^{\text{A,P}}$ from $\mathbb{X}_{i,k}^{\text{A,P}}$, for $\tau = 1, \dots, h$, as

$$\begin{aligned} \mathbb{X}_{i,k+\tau|k}^{\text{A,P}} &= \left(\mathbf{f}^t \left(\mathbb{X}_{i,k+\tau-1|k}^{\text{A,P}}, [\mathbf{v}^t] \right) \cap \mathbb{X}_g \right) \\ &\quad \setminus \left(\mathbb{X}_{i,k}^o \cup \left(\left(\mathbb{F} \left(\mathbf{x}_{i,k+\tau|k}^u \right) \cap \mathbb{X}_g \right) \setminus \mathbb{H}_{i,k}^{\text{CVS}} \right) \right). \end{aligned} \quad (5.18)$$

The main difference between (5.13) and (5.18) is that $\mathbb{X}_{i,k}^o$ is used in (5.18) to avoid exploring regions near obstacles because they are guaranteed to never contain a target location.

Then, as in Section 5.2.1, if some neighbors of UAV i with indexes in $\mathcal{N}_{i,k} \subset \mathcal{N}_{i,k}$ have already evaluated their sequence of control inputs $\hat{\mathbf{u}}_{\ell,k:h}$, $\ell \in \mathcal{N}_{i,k}$ and broadcast $\hat{\mathbf{x}}_{\ell,k:h|k}^u$ as well as $\mathbb{H}_{\ell,k}^{\text{CVS}}$, then (5.18) becomes

$$\begin{aligned} \mathbb{X}_{i,k+\tau|k}^{\text{A,P}} &= \left(\mathbf{f}^t \left(\mathbb{X}_{i,k+\tau-1|k}^{\text{A,P}}, [\mathbf{v}^t] \right) \cap \mathbb{X}_g \right) \\ &\quad \setminus \left(\mathbb{X}_{i,k}^o \cup \left(\left(\mathbb{F} \left(\mathbf{x}_{i,k+\tau|k}^u \right) \cap \mathbb{X}_g \right) \setminus \mathbb{H}_{i,k}^{\text{CVS}} \right) \cup \right. \\ &\quad \left. \bigcup_{\ell \in \mathcal{N}_{i,k}} \left(\left(\mathbb{F} \left(\hat{\mathbf{x}}_{\ell,k+\tau|k}^u \right) \cap \mathbb{X}_g \right) \setminus \mathbb{H}_{\ell,k}^{\text{CVS}} \right) \right). \end{aligned} \quad (5.19)$$

A drawback of the simplification $\mathbb{H}_{i,k+\tau|k}^{\text{CVS}} = \mathbb{H}_{i,k}^{\text{CVS}}$, with $\tau = 1, \dots, h$, is that it does not favor trajectories that turn around obstacles, as illustrated in the left part of Figure 5.2. Consider that the red UAV represents UAV i at time t_k . In this figure, the set $\mathbb{H}_{i,k}^{\text{CVS}}$ (delimited by a red hatched set) represents a coarse estimate of the portion of the ground that will not be observed in the future h time steps. Thus,

by following the trajectory represented by a blue curve, the next location of UAV i (the purple UAV) will allow an observation of what is behind the obstacle detected at time t_k . Nevertheless, due to the proposed simplification, the portion of $\mathbb{X}_{i,k+\tau|k}^{A,P}$ (in yellow) within $\mathbb{H}_{i,k}^{CVS}$ cannot be removed because it is assumed to remain hidden behind an obstacle, which is false. Thus, $\mathbb{X}_{i,k+h|k}^{A,P}$ is not as reduced as it should be. Moreover, as $\mathbb{H}_{i,k}^{CVS}$ only considers the obstacles detected at time t_k , the obstacles that are not within the FoV of UAV i at time t_k cannot be considered. Thus, since their shapes and locations are unknown, we still have the same problem as in Section 5.2.1 where, without knowledge of the map, the reduction of the FoV due to the unknown obstacles cannot be predicted. Consequently, the predicted reduction of the size of the set $\mathbb{X}_{i,k}^A$ is difficult to predict and trajectories which may lead to reductions will not necessarily be selected. This is illustrated on the right part of Figure 5.2 where the blue UAVs are two possible future locations of UAV i . From these locations, and since we assume $\mathbb{H}_{i,k+\tau|k}^{CVS} = \mathbb{H}_{i,k}^{CVS}$, the light-orange sets are portions of $\mathbb{X}_{i,k+h|k}^{A,P}$ that are removed during the prediction step. Nevertheless, this reduction will not happen because of the presence of unknown obstacles where the portion of the ground they will occlude, *i.e.*, $\mathbb{H}_{i,k+h}^{CVS}$, is represented by the purple hatched sets.

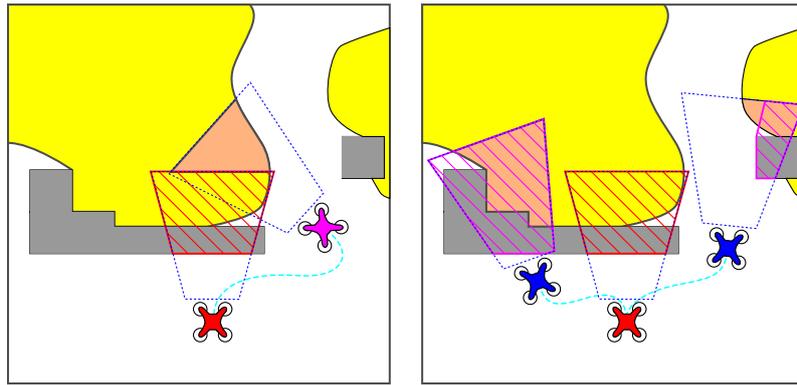


Figure 5.2 – Impact of the simplification $\mathbb{H}_{i,k+\tau|k}^{CVS} = \mathbb{H}_{i,k}^{CVS}$ (red hatched set), for $\tau = 1, \dots, h$ in the MPC trajectory design : the UAV location at time t_k is in red, the candidate trajectories and final locations are in blue or purple. Trajectories such that the FoV intersects $\mathbb{H}_{i,k}^{CVS}$ (left part) will be avoided due to the reduction of the usable part of the FoV in favor of trajectories without intersection between the FoV and $\mathbb{H}_{i,k}^{CVS}$ (right part). The purple hatched sets are the $\mathbb{H}_{i,k+h}^{CVS}$ obtained at each location represented by the blue UAVs.

Second criterion

As in Section 5.2.1, to address situations where all sequences of control inputs do not lead to any reduction of the localization uncertainty, a variant of J_i^{1a} introduced in (5.15) is proposed to account for $\mathbb{H}_{i,k}^{CVS}$:

$$J_i^{1b}(\mathbf{u}_{k:h}) = d\left(c\left(\mathbb{F}\left(\mathbf{x}_{i,k+h|k}^u\right) \cap \mathbb{X}_g\right), \mathbb{X}_{i,k+h|k}^{A,P} \setminus \mathbb{H}_{i,k}^{CVS}\right). \quad (5.20)$$

This avoids leading UAV i towards areas that are likely to be occluded by an obstacle. Then, combining (5.6) and (5.20), the second considered strategy aims at minimizing the criterion

$$J_i^{\text{STR}2}(\mathbf{u}_{k:h}) = J_i^0(\mathbf{u}_{k:h}) + \lambda J_i^{1b}(\mathbf{u}_{k:h}). \quad (5.21)$$

Compared to (5.16), (5.21) is computationally more demanding since (5.21) accounts for $\mathbb{H}_{i,k}^{\text{CVS}}$ and $\mathbb{X}_{i,k}^{\text{O}}$, even if both sets are constant during the evaluation.

5.2.3 . Exploiting the OEM

In this section, the OEM managed by each UAV is exploited to predict the parts of the ground that will be hidden by obstacles. The aim is to favor trajectories that observe areas that may contain a target while avoiding occlusions.

At time t_k , the OEM of UAV i $\mathcal{M}_{i,k}$ contains information regarding the location and the shape of the obstacles already observed. As in Section 5.2.2, for a trajectory leading to the exploration of partly uncharted parts of the environment, UAV i is unable to determine whether new obstacles will be observed. Consequently, in the MPC approach, the predicted map $\mathcal{M}_{i,k+\tau|k}$ is taken equal to $\mathcal{M}_{i,k}$ for all $\tau = 1, \dots, h$. The main difference with Section 5.2.2 is that here the part of the environment that will be hidden by obstacles can be predicted, even if $\mathcal{M}_{i,k}$ is not updated during the prediction.

The OEM $\mathcal{M}_{i,k}$ introduced in Section 4.6, can be used to define a list of boxes (4.41), *i.e.*,

$$\{\mathbb{B}_{i,k}^{\text{M}}(\eta) \mid s_{i,k}(\eta) = \{\text{O}\}, \eta \in \mathcal{N}^{\text{M}}\} \quad (5.22)$$

that approximates the shape of all obstacles that have been detected up to time t_k . Using these boxes, UAV i , with state \mathbf{x}_i^{u} , is able to evaluate the set $\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_i^{\text{u}})$ representing the portion of the ground hidden by all boxes within (4.41), and $(\mathbb{F}(\mathbf{x}_i^{\text{u}}) \cap \mathbb{X}_{\text{g}}) \cap \mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_i^{\text{u}})$ is the portion of $\mathbb{F}(\mathbf{x}_i^{\text{u}}) \cap \mathbb{X}_{\text{g}}$ that is predicted to be hidden by obstacles, see Section 4.6.4. Similarly, to predict for any $\tau = 1, \dots, h$ the portion of the ground that will be hidden by obstacle, *i.e.*, $\mathbb{P}_{i,k+\tau}^{\text{g}}(\mathcal{Y}_{i,k+\tau}^{\text{g}})$, UAV i evaluates $(\mathbb{F}(\mathbf{x}_{i,k+\tau|k}^{\text{u}}) \cap \mathbb{X}_{\text{g}}) \cap \mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_{i,k+\tau|k}^{\text{u}})$.

Therefore, (5.8) becomes

$$\mathbb{P}_{i,k+\tau|k}^{\text{g}}(\mathcal{Y}_{i,k+\tau|k}^{\text{g}}) = (\mathbb{F}(\mathbf{x}_{i,k+\tau|k}^{\text{u}}) \cap \mathbb{X}_{\text{g}}) \setminus \mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_{i,k+\tau|k}^{\text{u}}). \quad (5.23)$$

Then, as for (5.18), the equation (4.24) is used to predict the evolution of the targets and is corrected by an adaption of (4.30). This adaption accounts for the simplifications : $\mathcal{D}_{i,k+\tau}^{\text{t}} = \emptyset$, $\mathbb{X}_{i,k+\tau|k}^{\text{O}} = \mathbb{X}_{i,k}^{\text{O}}$, and (5.23). Consequently, UAV i can adapt (5.13) to evaluate iteratively $\mathbb{X}_{i,k+\tau|k}^{\text{A,P}}$ from $\mathbb{X}_{i,k}^{\text{A,P}}$, for $\tau = 1, \dots, h$, as

$$\begin{aligned} \mathbb{X}_{i,k+\tau|k}^{\text{A,P}} &= \left(\mathbf{f}^{\text{t}} \left(\mathbb{X}_{i,k+\tau-1|k}^{\text{A,P}}, [\mathbf{v}^{\text{t}}] \right) \cap \mathbb{X}_{\text{g}} \right) \\ &\quad \setminus \left(\mathbb{X}_{i,k}^{\text{O}} \cup \left(\mathbb{F}(\mathbf{x}_{i,k+\tau|k}^{\text{u}}) \setminus \mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_{i,k+\tau|k}^{\text{u}}) \right) \right). \end{aligned} \quad (5.24)$$

As in Section 5.2.1, if a subset of neighbors $\underline{\mathcal{N}}_{i,k} \subset \mathcal{N}_{i,k}$ of UAV i has already evaluated their sequence of control inputs $\hat{\mathbf{u}}_{\ell,k:h}$, with $\ell \in \underline{\mathcal{N}}_{i,k}$ and broadcast their sequence of states $\hat{\mathbf{x}}_{\ell,k:h|k}^{\text{u}}$, then,

assuming they share the same OEM, UAV i is able to evaluate their $\mathbb{H}_{\ell,k}^{\text{OEM}}(\hat{\mathbf{x}}_{\ell,k+\tau|k}^u)$, with $\ell \in \mathcal{N}_{i,k}$. Therefore, (5.18) becomes

$$\begin{aligned} \mathbb{X}_{i,k+\tau|k}^{\text{A,P}} &= \left(\mathbf{f}^t \left(\mathbb{X}_{i,k+\tau-1|k}^{\text{A,P}}, [\mathbf{v}^t] \right) \cap \mathbb{X}_g \right) \\ &\quad \setminus \left(\mathbb{X}_{i,k}^{\text{O}} \cup \left(\mathbb{F} \left(\mathbf{x}_{i,k+\tau|k}^u \right) \setminus \mathbb{H}_{i,k}^{\text{OEM}} \left(\mathbf{x}_{i,k+\tau|k}^u \right) \right) \right) \cup \\ &\quad \bigcup_{\ell \in \mathcal{N}_{i,k}} \left(\mathbb{F} \left(\hat{\mathbf{x}}_{\ell,k+\tau|k}^u \right) \setminus \mathbb{H}_{i,k}^{\text{OEM}} \left(\hat{\mathbf{x}}_{\ell,k+\tau|k}^u \right) \right). \end{aligned} \quad (5.25)$$

Third criterion

As in Section 5.2.2, to address situations where all sequences of control inputs do not lead to any reduction of the localization uncertainty, a variant of J_i^{1a} in (5.15) is introduced to account for $\mathbb{H}_{i,k}^{\text{OEM}}$, *i.e.*,

$$J_i^{1c}(\mathbf{u}_{k:h}) = d \left(c \left(\mathbb{F} \left(\mathbf{x}_{i,k+h|k}^u \right) \cap \mathbb{X}_g \right), \mathbb{X}_{i,k+h|k}^{\text{A,P}} \setminus \mathbb{H}_{i,k}^{\text{OEM}} \left(\mathbf{x}_{i,k+h|k}^u \right) \right). \quad (5.26)$$

This avoids trajectories leading UAV i towards parts of the environment that are hidden by an obstacle. Therefore, by combining (5.6) and (5.26) the following criterion can be defined

$$J_i^{\text{STR3}}(\mathbf{u}_{k:h}) = J_i^0(\mathbf{u}_{k:h}) + \lambda J_i^{1c}(\mathbf{u}_{k:h}). \quad (5.27)$$

Compared to (5.21), (5.27) is more computationally demanding since the UAV has to evaluate $\mathbb{H}_{i,k}^{\text{OEM}}$ for each prediction-correction process (5.25).

5.2.4 . Promoting the exploration of recently hidden areas

In (5.27), J_i^0 favors trajectories leading to future reductions of the size of $\mathbb{X}_{i,k}^{\text{A}}$, whereas J_i^{1c} is used to drive the UAV closer to the set estimates. Neither J_i^0 nor J_i^{1c} in (5.27) favor trajectories leading to the exploration of parts of $\mathbb{X}_{i,k}^{\text{A}}$ that were previously hidden. This section introduces a term $J_i^2(\mathbf{u}_{k:h})$ to complete (5.27) to favor such trajectories and search for previously potentially hidden targets.

Using the approximation of the hidden portion of the ground deduced from CVS information $\mathbb{H}_{i,k}^{\text{CVS}}$, each UAV is able to evaluate the portion of $\mathbb{X}_{i,k}^{\text{A}}$ that has not been observed, *i.e.*, $\mathbb{X}_{i,k}^{\text{A}} \cap \mathbb{H}_{i,k}^{\text{CVS}}$. Then, consider the hidden parts of the RoI for UAV i and its neighbors at time t_k

$$\overline{\mathbb{H}}_{i,k}^{\text{CVS}} = \bigcup_{\ell \in \mathcal{N}_{i,k}} \mathbb{H}_{\ell,k}^{\text{CVS}}, \quad (5.28)$$

and the surface of $\mathbb{X}_{i,k+h|k}^{\text{A,P}}$ that may be hidden

$$J_i^2(\mathbf{u}_{k:h}) = \phi \left(\mathbb{X}_{i,k+h|k}^{\text{A,P}} \cap \overline{\mathbb{H}}_{i,k}^{\text{CVS}} \right). \quad (5.29)$$

By introducing (5.29) in (5.27), UAV i can search for trajectories leading to a reduction of the size of $\mathbb{X}_{i,k+h|k}^{\text{A,P}} \cap \overline{\mathbb{H}}_{i,k}^{\text{CVS}}$. Therefore, if $\mathbb{H}_{i,k}^{\text{CVS}} = \emptyset$, then one remarks that UAV i can search for trajectories leading to the exploration of set estimates that were previously hidden for its neighbors. Thus, the term (5.29) implies a cooperative behavior because two UAVs may explore what the other has not been able to observe.

Nevertheless, when UAV i designs a trajectory to explore over the time interval $[t_{k+1}, t_{k+h}]$ what was hidden at time t_k , it also has to take into account the portion of $\mathbb{X}_{i,k}^A$ that was previously hidden but has not been explored yet. To explain why it is important to take into account for the $\overline{\mathbb{H}}_{i,k}^{\text{CVS}}$ obtained at a previous time instant, we consider the following example. Let assume that at time t_{k-1} , UAV i designed a trajectory to explore $\mathbb{X}_{i,k-1+h}^{\text{A,P}} \cap \overline{\mathbb{H}}_{i,k-1}^{\text{CVS}}$, but the UAV will start the exploration of that area only at time t_{k-1+h} . Then, during the trajectory update at time t_k , if $\overline{\mathbb{H}}_{i,k}^{\text{CVS}} = \emptyset$, then, according to (5.29) $J_i^2(\mathbf{u}_{k:h}) = 0$ regardless of the sequence $\mathbf{u}_{k:h}$. Consequently, UAV i forgets that it was supposed to explore $\mathbb{X}_{i,k-1+h}^{\text{A,P}} \cap \overline{\mathbb{H}}_{i,k-1}^{\text{CVS}}$.

To avoid this issue, we consider the union

$$\overline{\mathbb{H}}_{i,k-h+1:h}^{\text{CVS}} = \bigcup_{t=k-h+1}^k \overline{\mathbb{H}}_{i,t}^{\text{CVS}} \quad (5.30)$$

of the h past $\overline{\mathbb{H}}_{i,t}^{\text{CVS}}$. Using $\overline{\mathbb{H}}_{i,k-h+1:h}^{\text{CVS}}$ instead of $\overline{\mathbb{H}}_{i,k}^{\text{CVS}}$, (5.29) becomes

$$J_i^2(\mathbf{u}_{k:h}) = \phi \left(\mathbb{X}_{i,k+h|k}^{\text{A,P}} \cap \overline{\mathbb{H}}_{i,k-h+1:h}^{\text{CVS}} \right). \quad (5.31)$$

Consequently, by combining (5.6), (5.26), and (5.31) to get the criterion

$$J_i^{\text{STR3}}(\mathbf{u}_{k:h}) = J_i^0(\mathbf{u}_{k:h}) + \lambda J_i^{1c}(\mathbf{u}_{k:h}) + \kappa J_i^2(\mathbf{u}_{k:h}), \quad (5.32)$$

UAV i can favor trajectories leading to the exploration of recently hidden areas that may contain hidden targets.

The value of κ represents a trade-off between the general target search and the exploration of specific areas that may contain hidden targets. This value may depend on the density of the obstacles within the RoI, on their structure, and how the UAVs are able to monitor an area.

5.3 . MPC implementation

This section presents how we implemented the MPC on the UAVs. Section 5.3.1 describes some design choices in the MPC approach such as the determination of the prediction horizon h and how to simplify the evaluation of $\mathbb{X}_{i,k+\tau|k}^{\text{A,P}}$ for $\tau = 1, \dots, h$. Then, Section 5.3.2 details the organization of computations and communications.

5.3.1 . Practical choices

Assuming that the UAVs are flying at a fixed height and with a constant speed, then the sequence of control inputs $\hat{\mathbf{u}}_{i,k:h}$ provided by the MPC to the low-level controller is chosen to be a sequence of yaw angle increments. Each control input $\hat{\mathbf{u}}_{i,k+\tau} \in \hat{\mathbf{u}}_{i,k:h}$, for $\tau = 0, \dots, h-1$, is taken from a set of feasible control inputs $\mathbb{U} = [-\Delta\psi_{\max}, \Delta\psi_{\max}]$ where $\Delta\psi_{\max}$ is the considered maximal angle increment.

To determine the period T^{MPC} to update the control input with the MPC, we consider the time response of the quadcopter regarding the evolution of its yaw angle. A step response for a step of $\pi/3$ radian is used to evaluate this time response, see Figure 5.3. By approximating the step response

to a second-order system, one remarks that the system needs $t_{95\%} = 1.2$ seconds to reach a steady-state error of less than 5 %. From that measure, we consider that the quadcopter requires $T^{\text{MPC}} \approx 3t_{95\%}$ to reach the desired angle. Thus, we take $T^{\text{MPC}} = 3$ seconds.

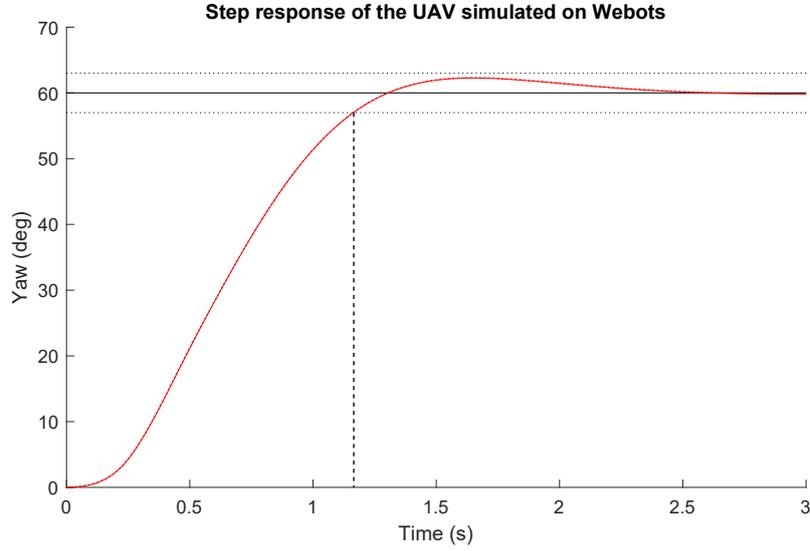


Figure 5.3 – Evolution of the yaw angle of a UAV in response to a yaw angle step of $\pi/3$ radian (or 60 degrees).

Then, consider the trajectory $\hat{x}_{i,k:h|k}^u$ of UAV i , updated at time t_k . This trajectory has been designed to guide UAV i over the time window $[t_{k+1}, t_{k+h}]$. A trajectory designed by the MPC at some time instant t_k should be designed to guide the UAV until it is re-updated, *i.e.*, at time $t_k + T^{\text{MPC}}$. Consequently, we must have

$$t_k + T^{\text{MPC}} \leq t_{k+h} \Leftrightarrow t_k + T^{\text{MPC}} \leq t_k + hT \quad (5.33)$$

where T is representative of the time needed for the target location estimator to update the set estimates from the CVS measurements. Therefore, one has $h \geq T^{\text{MPC}}/T$. Besides, since h is an integer, one has instead $h \geq \text{ceil}(T^{\text{MPC}}/T)$, where $\text{ceil}(x)$ rounds x to the nearest integer greater than or equal to x . In what follows, we use $h' = \text{ceil}(T^{\text{MPC}}/T)$ as the smallest possible value that can take the prediction horizon h .

The value of h can be chosen as high as possible as long as $h \geq h'$. The higher h is, the more capable UAV i is to design a trajectory that will efficiently reduce the localization uncertainty, but the heavier the computational burden will be, see Figure 5.4. So, one may choose the prediction horizon h such that the trajectory $\hat{x}_{i,k:h|k}^u$ is designed to last for $N^{\text{MPC}} \geq 1$ multiple calls of the MPC, *i.e.*, $h = N^{\text{MPC}}h'$.

Figure 5.4 illustrates two cases of trajectory update, both with $h' = 3$. The left part illustrates the trajectory update with $N^{\text{MPC}} = 2$, whereas the right part illustrates a trajectory update with $N^{\text{MPC}} = 1$. Both trajectories explore the green set, but, with $N^{\text{MPC}} = 2$, UAV i is able to design a trajectory that will lead to a quicker exploration of the yellow set, since the UAV predicts that at time t_{k+h} ($h = 6$ here) it will start its exploration.

We assume that a UAV requires T^{MPC} to reach the desired angle for a given control input. Therefore, if $T^{\text{MPC}}/T > 1$ and since the control inputs are angle increments, then the sequence of

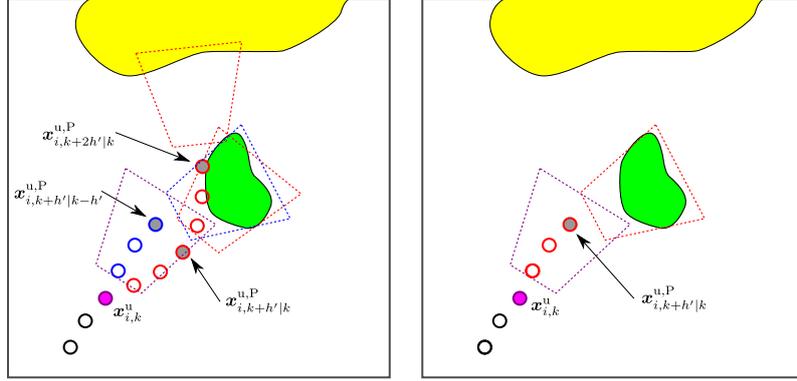


Figure 5.4 – Illustration of two trajectory updates, one with $N^{\text{MPC}} = 2$ (left), the other with $N^{\text{MPC}} = 1$ (right), while having $h' = 3$. The purple circle is the current location of UAV i (and also $\hat{x}_{i,k|k-h'}^u$). The blue and black circles are the locations of UAV i according to the sequence of control inputs obtained at time $t_{k-h'}$ whereas the red circles are the locations of UAV i according to the sequence obtained at time t_k . The difference between the blue and black circles are that the black circles are actually representing the trajectory that UAV i has taken, whereas the blue circles are what remains of the previous trajectory, *i.e.*, the fragment of $\hat{x}_{i,k-h':h|k-h'}$ before the trajectory update at time t_k . For the sake of simplicity, the targets are not moving here.

control inputs $(\hat{\mathbf{u}}_{i,k}, \dots, \hat{\mathbf{u}}_{i,k+h'-1})$ is such that $\hat{\mathbf{u}}_{i,k} = \hat{\mathbf{u}}_{i,k}$ and $\hat{\mathbf{u}}_{i,k+\tau} = 0$ with $\tau = 1, \dots, h' - 1$, *i.e.*, the desired angle must not change until $t_{k+h'}$. The same reasoning is applied for each fragment $(\hat{\mathbf{u}}_{i,k+nh'}, \dots, \hat{\mathbf{u}}_{i,k+(n+1)h'-1})$ of the sequence of control inputs $\hat{\mathbf{u}}_{i,k:h}$, with $n = 0, \dots, N^{\text{MPC}} - 1$.

Consider that the set of feasible control inputs is finite. Then an approach to search for the optimal trajectory is to test all combinations of control input sequences, *i.e.*, by a tree search. This implies a computational burden that will slow the MPC.

We have assumed that the time T needed for the target location estimator to update the set estimates is much smaller than the time T^{MPC} needed for the UAV to reach the desired state. So, one can assume that the predicted set estimates do not really change between two consecutive time instants $t_{k+\tau}$ and $t_{k+\tau+1}$. Thus, one can lighten the computational burden of the MPC by making the prediction of the evolution of $\mathbb{X}_{i,k+\tau|k}^{\text{A,P}}$ T^{MPC} -periodic instead of T -periodic.

Nevertheless, while this simplification reduces the computational burden required to evaluate all possible combinations of the sequence of control inputs, it also reduces the quality of the predicted set estimates and, thus, the predicted reduction of the localization uncertainty. Figure 5.5 illustrates the pros and cons of the proposed simplification with $N^{\text{MPC}} = 2$ and $T^{\text{MPC}}/T = 3$.

5.3.2 . Task scheduling

Another problem to consider is when the neighbors of UAV i broadcasts their trajectory and other relevant information.

A first approach is to consider a sequential scheduling where the UAV in the neighborhood of UAV i with the lower index starts the trajectory update. Once finished, it broadcasts its updated trajectory, and the UAV with the second lower index searches for the best sequence of control inputs that minimizes one of the aforementioned criteria, *i.e.*, (5.16), (5.21), or (5.32), while accounting for

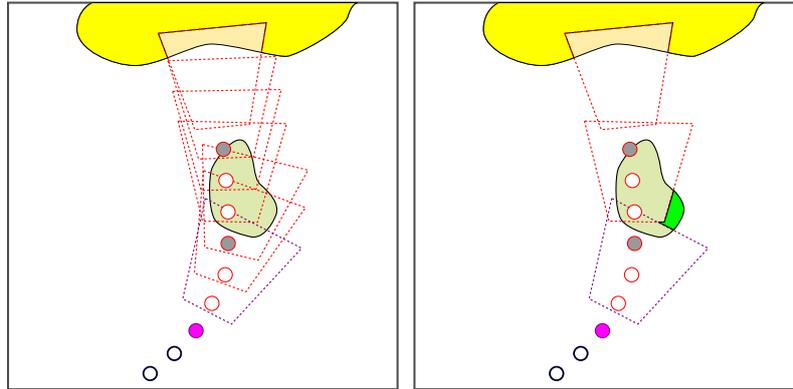


Figure 5.5 – Comparison between an approach where the prediction of the set estimates is done with a period T (right) and (left) an approach where the set estimates are only updated with a period $T^{\text{MPC}} = 3T$. The light green and light yellow part of the set estimates represent the observed parts. Targets are assumed static.

the trajectory of its predecessor. This scheme is repeated until each UAV has updated its trajectory. Figure 5.6 illustrates this sequential scheduling with 3 UAVs.

A drawback of this approach is that the UAV with the highest index has to wait for the others before updating its trajectory, and this delay may be larger than T . Another issue is illustrated in Figure 5.6. The updated trajectory of UAV 1 is not cooperative since it does not account for any of the trajectories of its neighbors. A possible approach to solve this issue is that UAV 1 waits to receive the trajectory of UAV 2 and 3 before re-evaluating the best sequence of control inputs. This re-evaluation is performed by UAV 2 and 3 as well. This loopback is likely to improve the cooperation between UAVs, but at the cost of doubling the time needed without loop.

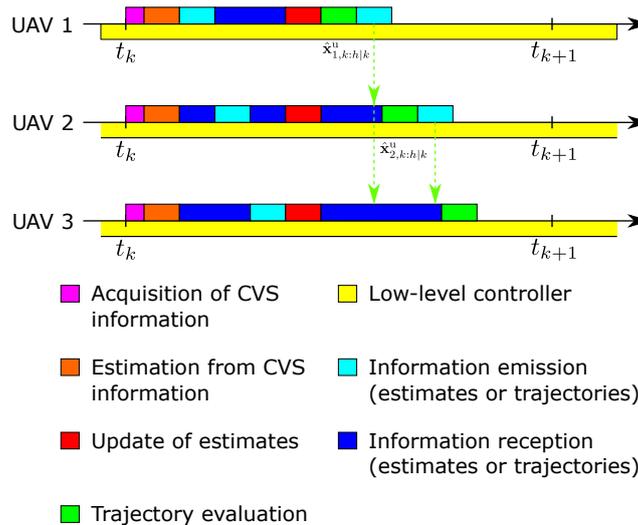


Figure 5.6 – Succession of tasks performed by three UAVs; here the trajectory update with the MPC is performed sequentially.

Another approach is to consider a non-sequential scheduling where the UAVs only work with the previous trajectory of their neighbors. At time t_k , UAV i keeps in its memory the sequence of states $\hat{\mathbf{x}}_{\ell, k-h':h|k-h'}^u$ of the neighbors it had at time $t_{k-h'}$, i.e., $\ell \in \mathcal{N}_{i, k-h'}$. For a $\ell \in \mathcal{N}_{i, k-h'}$, one can remark that only the portion $(\hat{\mathbf{x}}_{\ell, k+1|k-h'}^u, \dots, \hat{\mathbf{x}}_{\ell, k+h-h'|k-h'}^u)$ of $\hat{\mathbf{x}}_{\ell, k-h':h|k-h'}^u$ is useful. This is because all predicted states $\hat{\mathbf{x}}_{\ell, k-h'+\tau|k-h'}^u$ with $\tau = 0, \dots, h' - 1$ are actually in the past.

UAV i can use $(\hat{\mathbf{x}}_{\ell, k+1|k-h'}^u, \dots, \hat{\mathbf{x}}_{\ell, k+h-h'|k-h'}^u)$ as an estimate of the trajectory that UAV ℓ is supposed to take. Therefore, one has $\hat{\mathbf{x}}_{\ell, k+\tau|k}^u = \hat{\mathbf{x}}_{\ell, k+\tau|k-h'}^u$ for $\tau = 0, \dots, h - h'$. One can remark that this approach only works if we have $N^{\text{MPC}} > 1$. Then, for the future time instants $t_{k+h-h'+\tau}$ with $\tau = 1, \dots, h'$, a possible assumption is to consider that UAV ℓ only flies in a straight line while maintaining the same orientation as at time $t_{k+h-h'}$.

The main advantage of the latter approach is that no UAV has to wait for its neighbors to update its trajectory.

Figure 5.7 illustrates this non-sequential approach.

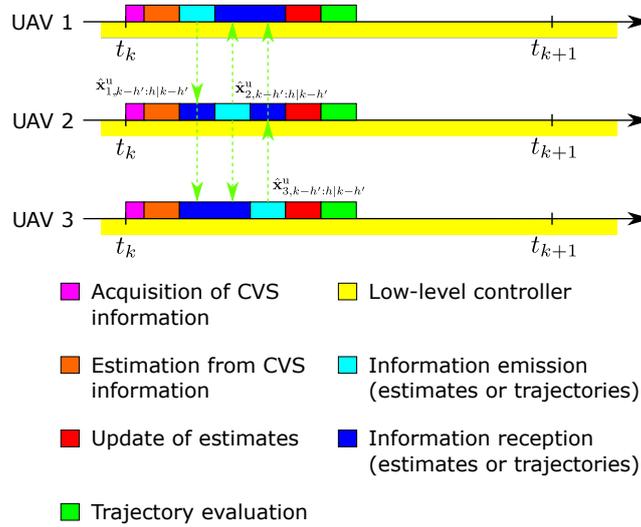


Figure 5.7 – Succession of tasks performed by three UAVs; here the trajectory update with the MPC is performed using only the previously known trajectories of the UAV's neighbors

5.4 . Simulation

The section presents simulation results to compare the performance of the criteria (5.16), (5.21), or (5.32) designed to update the UAV trajectory to solve a CSAT problem within an unknown and cluttered environment. Section 5.4.1 introduces the simulation conditions. Then, Section 5.4.2 describes the metrics used to compare the control strategies. Section 5.4.3 compares and analyses the criteria defined for target search and the minimization of the localization uncertainty. Section 5.4.4 focuses only on the exploration, and Section 5.4.5 provides some results regarding the computing time of the estimator and the MPC.

5.4.1 . Simulation parameters

Simulations have been performed using a combination of Webots ([Webots](#),) to generate UAV and target displacements as well as the measurements collected by the UAVs, and MATLAB where the estimation and trajectory design algorithms have been implemented. The communication between Webots and MATLAB is done via TCP-IP.

The RoI is a simplified urban environment with different types of buildings, all with a height less than 50 m, see Figure 5.8. One has $\mathbb{X}_o = [-250 \text{ m}, 250 \text{ m}] \times [-250 \text{ m}, 250 \text{ m}] \times \mathbb{R}^+$. The ground area occupied by building represents 5 % of the total area of \mathbb{X}_g . The RoI contains $N^t = 8$ targets (cars of the same shape).

The resolution of the occupancy elevation map $\mathcal{M}_{i,k}$ for all UAVs is taken as 5 m.

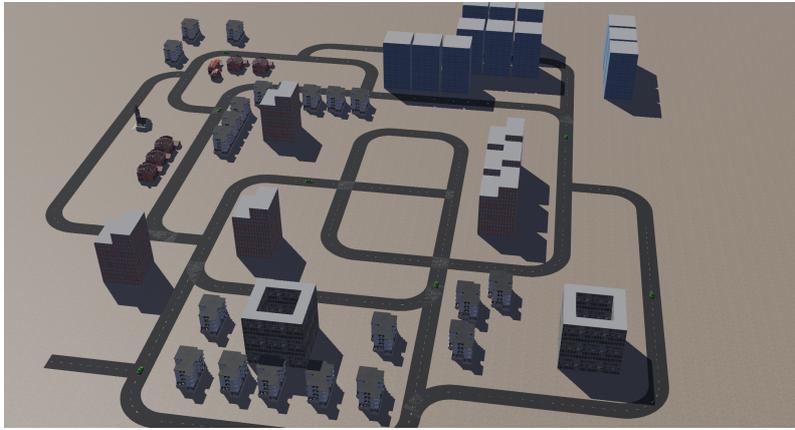


Figure 5.8 – Simulated urban environment in Webots

Each car has the following sizes $4.6 \text{ m} \times 1.8 \text{ m} \times 1.5 \text{ m}$, and can be included in a cylinder \mathbb{C}^t of radius $r^t = 2.5 \text{ m}$ and height $h^t = 2 \text{ m}$. The safety distances are taken as $r_s^{to} = 2.5 \text{ m}$ and $r_s^{tt} = 8 \text{ m}$. The state $\mathbf{x}_{j,k}^t = (x_{j,1,k}^t, x_{j,2,k}^t, x_{j,3,k}^t)$ of target $j \in \mathcal{N}^t$ consists of the coordinates $(x_{j,1,k}^t, x_{j,2,k}^t)$ of the projection on \mathbb{X}_g of its center of mass, and of its heading angle $x_{j,3,k}^t$. Its dynamic model, *c.f.*, 3.2, is

$$\begin{pmatrix} x_{j,1,k+1}^t \\ x_{j,2,k+1}^t \\ x_{j,3,k+1}^t \end{pmatrix} = \begin{pmatrix} x_{j,1,k}^t + T v_{j,1}^t \cos(x_{j,3,k}^t) \\ x_{j,2,k}^t + T v_{j,1}^t \sin(x_{j,3,k}^t) \\ x_{j,3,k}^t + T v_{j,2,k}^t \end{pmatrix}, \quad (5.34)$$

where $v_{j,1}^t$ is the constant target speed and $v_{j,2,k}^t$ is the target turn rate. At the start of the simulation $v_{j,1}^t$ is randomly chosen in the interval $[0, v_{\max}]$ for each target $j \in \mathcal{N}^t$. At each time instant, $v_{j,2,k}^t$ is constrained to maintain the car on the roads. The UAVs do not exploit any knowledge related to the roads during the search.

A fleet of 4 identical quadcopters is considered (the DJI Mavic2Pro model in Webots). As in Section 4.7 of Chapter 4, these UAVs are equipped with a camera of resolution $N_r \times N_c = 720 \times 960$ pixels, but the processed image is downsampled to work with a resolution of $N_r \times N_c = 360 \times 480$ pixels, aperture angle of $\pi/4$ rad, and orientation in the UAV body frame of $\theta = \pi/6$ rad. A depth map is provi-

ded by the *range finder* of Webots with the same resolution as the camera. Its maximal measurement range is set to $d_{\max} = 300$ m and the bounds of the noise w in (3.33) are $[\underline{w}, \bar{w}] = [-0.01, 0.01]$.

The CVS provides measurements with a period $T = 0.5$ s. The set estimates are updated with the same period using the algorithms presented in Chapter 4. Since only $v_{\max} = 1$ m/s is known to the UAVs, the model of the dynamics of the targets used to perform the prediction is (5.34) but we replace $v_{j,1}^t$ by v_{\max} . Actually, since the UAVs only know v_{\max} , then the prediction of the target-related set estimates is simply a dilatation.

Regarding the MPC, the value of the prediction horizon is $h = 12$, i.e., $N^{\text{MPC}} = 2..$ Moreover, instead of using the quadcopter dynamics, see (Bouabdallah, 2007), during the UAV trajectory update, a simpler dynamic model with constant altitude and speed v^u , as well as null roll and pitch angles are considered

$$\begin{pmatrix} x_{i,1,k+1}^u \\ x_{i,2,k+1}^u \\ x_{i,3,k+1}^u \\ x_{i,4,k+1}^u \end{pmatrix} = \begin{pmatrix} x_{i,1,k}^u + Tv^u \cos(x_{i,4,k}^u + u_{i,k}) \\ x_{i,2,k}^u + Tv^u \sin(x_{i,4,k}^u + u_{i,k}) \\ x_{i,3,k+1}^u \\ x_{i,4,k}^u + u_{i,k} \end{pmatrix}. \quad (5.35)$$

A low-level PID controller is implemented on each UAV to ensure a constant speed of $v^u = 5$ m/s and a constant fly height. This fly height guarantees that the UAVs fly above the obstacles. Moreover, the fly height of each UAV is different from the others to avoid collisions.

All the components of $\mathbf{u}_{k:h}$ are yaw angle increments taken in $\mathbb{U} = \{-\pi/3, -\pi/6, 0, \pi/6, \pi/3\}$. As in (Ibenthal et al., 2021), the first weight parameter of the criteria (5.16), (5.21), or (5.32) is taken as $\lambda = 10^{-5}$. The second weight parameter of the criterion (5.32) is set to $\kappa = 1$.

No restriction is considered on the communication range between the UAVs, and we assume that the communications are done without error or delay. In addition, to organize the computations and communications between the UAVs, we consider the sequential scheduling, see Figure 5.6, with a loopback.

All UAVs are launched from the same location outside the Rol.

5.4.2 . Evaluation metrics

To evaluate the performance of the propose approach regarding the accuracy of the set estimates of identified targets, one considers the average localization error with respect to a point estimate taken as the barycenter $c(\mathbb{X}_{i,j,k}^t)$ of the set estimate

$$e_{i,k}^t = \frac{1}{\text{card}(\mathcal{L}_{i,k}^t)} \sum_{j \in \mathcal{L}_{i,k}^t} \left\| \mathbf{x}_{j,k}^{\text{tg}} - c(\mathbb{X}_{i,j,k}^t) \right\|, \quad (5.36)$$

as well as the average surface of the set estimates of identified targets

$$\Phi_{i,k}^t = \frac{1}{\text{card}(\mathcal{L}_{i,k}^t)} \sum_{j \in \mathcal{L}_{i,k}^t} \phi(\mathbb{X}_{i,j,k}^t), \quad (5.37)$$

and the average

$$\Phi_{i,k}^{\text{t,eR}} = \frac{1}{\text{card}(\mathcal{L}_{i,k}^t)} \sum_{j \in \mathcal{L}_{i,k}^t} \phi^{\text{eR}}(\mathbb{X}_{i,j,k}^t), \quad (5.38)$$

where $\phi^{\text{eR}}(\mathbb{X})$ is the radius of a disc of area $\phi(\mathbb{X})$, *i.e.*,

$$\phi^{\text{eR}}(\mathbb{X}) = \sqrt{\frac{\phi(\mathbb{X})}{\pi}}. \quad (5.39)$$

Since there is no restriction on the communication range between the UAVs, we have $\mathcal{N}_{i,k} = \mathcal{N}^u$, the UAVs share the same set estimates and, thus, the values of $e_{i,k}^t$, $\Phi_{i,k}^t$, and $\Phi_{i,k}^{\text{t,eR}}$ are the same for all UAVs $i \in \mathcal{N}^u$.

The theoretical coverage of the FoV of the UAVs at time k

$$\Phi_k^{\text{FoV}} = \phi \left(\bigcup_{i \in \mathcal{N}^u} (\mathbb{F}(\mathbf{x}_{i,k}^u) \cap \mathbb{X}_g) \right) \quad (5.40)$$

represents the part of \mathbb{X}_g observed at time t_k by all UAVs if there is no obstacle nor target. The part of the ground that is actually seen and identified as Ground at time t_k is

$$\Phi_k^g = \phi \left(\bigcup_{i \in \mathcal{N}^u} \mathbb{P}_i^g(\mathcal{Y}_{i,k}^g) \right). \quad (5.41)$$

The cumulated ground surface observed up to time t_k is

$$\Phi_k^{\text{Cg}} = \phi \left(\bigcup_{\tau=1}^k \bigcup_{i \in \mathcal{N}^u} \mathbb{P}_i^g(\mathcal{Y}_{i,\tau}^g) \right). \quad (5.42)$$

Most of these metrics may also be expressed relative to the ground surface $\phi(\mathbb{X}_g)$.

Additional metrics of interest are the surface $\phi(\overline{\mathbb{X}}_{i,k}^t)$ of the set $\overline{\mathbb{X}}_{i,k}^t$ and the surface $\phi(\overline{\mathbb{X}}_{i,k}^t \cap \mathbb{X}_{i,k}^h)$ of the part of $\overline{\mathbb{X}}_{i,k}^t$ which has been occluded by an obstacle. In this last metric, the set $\mathbb{X}_{i,k}^h$ represents the portion of \mathbb{X}_g that has not been seen by UAV i at a previous time instant due to an occlusion by an obstacle and has not yet been observed again up to time t_k . This set is initialized with $\mathbb{X}_{i,0}^h = \emptyset$ and is updated as

$$\mathbb{X}_{i,k|k}^h = \left(\mathbb{X}_{i,k-1}^h \cup \mathbb{H}_{i,k}^{\text{CVS}} \right) \setminus \mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g). \quad (5.43)$$

Moreover, using information received from its neighbors, UAV i updates $\mathbb{X}_{i,k|k}^h$ as

$$\mathbb{X}_{i,k}^h = \left(\bigcup_{\ell \in \mathcal{N}^u} \mathbb{X}_{\ell,k|k}^h \right) \setminus \left(\bigcup_{\ell \in \mathcal{N}^u} \mathbb{P}_{\ell,k}^g(\mathcal{Y}_{\ell,k}^g) \right). \quad (5.44)$$

To evaluate the quality of the prediction of the hidden area using the elevation map, *i.e.*, $\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_{i,k}^u)$, when compared with the actual portion of the ground that is hidden, *i.e.*, $\mathbb{H}_{i,k}^{\text{CVS}}$, one can use the metric (4.50) introduced in Section 4.7 of Chapter 4, *i.e.*, the intersection over union of $\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_{i,k}^u) \cap \mathbb{F}(\mathbf{x}_{i,k}^u)$ and $\mathbb{H}_{i,k}^{\text{CVS}}$:

$$\Phi_k^{\text{OEM}} = \frac{1}{\text{card}(\mathcal{N}^u)} \sum_{i \in \mathcal{N}^u} \frac{\phi \left(\mathbb{H}_{i,k}^{\text{CVS}} \cap \left(\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_{i,k}^u) \cap \mathbb{F}(\mathbf{x}_{i,k}^u) \right) \right)}{\phi \left(\mathbb{H}_{i,k}^{\text{CVS}} \cup \left(\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_{i,k}^u) \cap \mathbb{F}(\mathbf{x}_{i,k}^u) \right) \right)}. \quad (5.45)$$

5.4.3 . Target search and tracking performance

This section compares the strategies associated with the criteria (5.16), (5.21), (5.27), and (5.32). Their performance is evaluated during cooperative target search and tracking in the unknown environment illustrated in Figure 5.8. All of the following results have been averaged over 10 independent simulations, each lasting 300 s. Table 5.2 contains links to videos illustrating a typical evolution of the UAVs and their set estimates for each criterion.

Note that, in what follows, we compare (5.21) with (5.21) when \mathbb{H}^{CVS} is not considered. The purpose of this comparison is to see how the addition of \mathbb{H}^{CVS} can improve the target search and tracking. Thus, when \mathbb{H}^{CVS} is not considered, it is as if $\mathbb{H}_{\ell,k}^{CVS} = \emptyset$ in (5.19) and (5.20), with $\ell \in \underline{\mathcal{N}}_{i,k} \cup \{i\}$.

Criterion	Evolution set estimates	Mapping of the RoI
(5.16)	Video_Crit_1	-
(5.21) (no \mathbb{H}^{CVS})	Video_Crit_2_noH	-
(5.21)	Video_Crit_2	-
(5.27)	Video_Crit_3	OEM_Crit_3
(5.32)	Video_Crit_3_bis	OEM_Crit_3_bis

Table 5.2 – Videos illustrating a typical evolution of the UAVs during the target search. $\bar{\mathbb{X}}_{i,k}^t$ is the yellow set. $\mathbb{X}_{i,k}^o$ is the black sets. $\mathbb{X}_{i,k}^h$ is the blue sets. $\mathbb{X}_{i,j,k}^t$ is the green sets. $\mathbb{H}_{i,k}^{CVS}$ is shown in red.

Figure 5.9 shows the evolution of the number card $\left(\mathcal{L}_{i,k}^t\right)$ of identified targets (left figure) and the number of targets included in the FoV of at least one UAVs (right figure-black curve) with the number of targets that are actually identified, *i.e.*, card $\left(\bigcup_{\ell \in \mathcal{N}^u} \mathcal{D}_{\ell,k}^t\right)$ (right figure-green curve).

The UAVs, in most cases, manage to identify the 8 targets located within the RoI. With the criterion (5.27) (the magenta curve), the curve remains around 7.2 identified targets : in only 2 out of the 10 simulations, the fleet has managed to find the 8 targets, whereas the UAVs only found 7 targets during the 8 other simulations.

The UAVs do not know the exact number of targets to identify. Nonetheless, if we disregard the magenta curve, the UAVs manage to identify and localize all the targets in less than 150 seconds. With criterion (5.32) (the black curve), the UAVs manage to identify all targets in less than 100 seconds. The inclusion of the term (5.31) in the criterion which favors trajectories leading to the exploration of areas recently occluded and thus ensuring the exploration in the vicinity of a UAV, instead of a general exploration that leaves unchecked those recently occluded areas.

Regarding the right part of Figure 5.9, the difference between the black and the green curves can be explained by the fact that a target may be physically located within the FoV of a UAV but either too far from the UAV to be identified, or hidden behind an obstacle. There is a larger difference between the black and green curves when the UAV trajectory design is performed based on the criterion (5.16), than with the criterion (5.32). This is because, unlike criterion (5.32), criterion (5.16) does nothing to take into account the obstacles. With this criterion, the set estimate $\mathbb{X}_{i,k}^o$ (region around obstacles that is free of targets) is not characterized, and the OEM is not available. Consequently, when a UAV updates its trajectory, it cannot design a target search trajectory that minimizes the occlusions and

may even be attracted toward the obstacles. The reason is, according to the correction equations in Section 4.5.2 and (5.13), since we have $\mathbb{X}_{i,k}^o = \emptyset$ with criterion (5.16), then the portions of $\mathbb{X}_{i,k+\tau|k}^{A,P}$, for $\tau = 1, \dots, h$, that are too close to an obstacle (or inside an obstacle) are not removed. Consequently, with that criterion, the UAVs are more likely to have obstacles within their FoV and, thus, have targets hidden behind by those obstacles. In criterion (5.32), $\mathbb{X}_{i,k}^o$ is characterized and the OEM is exploited to predict the occluded in the future. Consequently, the UAVs are able to design target search trajectories that lead them towards regions of the RoI where the targets can actually be (they are not within the neighborhood of the obstacles) while minimizing the portion of $\mathbb{X}_{i,k+\tau|k}^{A,P}$, for $\tau = 1, \dots, h$ that is hidden behind the obstacles. Thus, with that criterion, the targets are less likely to be hidden behind an obstacle.

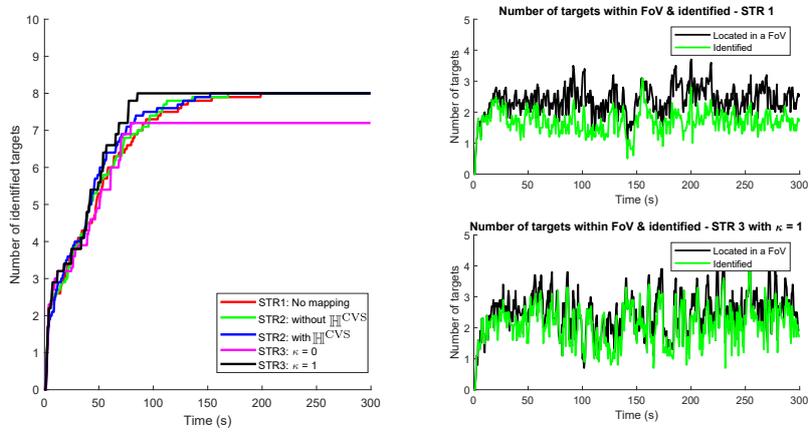


Figure 5.9 – (Left) Time evolution of card $(\mathcal{L}_{i,k}^t)$ for each considered criterion. (Right) Evolution of the number of targets located in the FoV of at least one UAV (black), and of the number of identified targets, *i.e.*, card $(\bigcup_{\ell \in \mathcal{N}^u} \mathcal{D}_{\ell,k}^t)$, (green). Only criterion (5.16) (top-right) (5.32) (bottom-right) are considered. These curves are average of 10 simulations.

Figure 5.10 illustrates the evolution of $e_{i,k}^t$, the evolution of $\Phi_{i,k}^t$, and the evolution of $\Phi_{i,k}^{t,eR}$ for each criterion. Table 5.3 gathers the average and the standard deviation of those metrics between $t = 150$ s and $t = 300$ s, for each criterion.

For all criteria, the average localization error is between 1 m and 1.5 m. Criterion (5.16) provides the worst results as obstacles are not taken into account. The criteria (5.27) and (5.32) provide the best results illustrating the benefit provided by the OEM in the trajectory design of the UAVs.

The top-right part of Figure 5.10 shows the evolution of the average localization uncertainty $\Phi_{i,k}^t$ evaluated as the surface of $\mathbb{X}_{i,j,k}^t$ averaged over each detected target, see (5.37). One has $\Phi_{i,k}^t$ about 0.3 % of the RoI area. The radius of a disc of equivalent surface is represented in Figure 5.10 (bottom-right). The obtained radius is between 12 m and 18 m depending of the criterion, showing that even if the localization error remains small, the uncertainty regarding the estimated location is large.

As shown in Figure 5.9, the UAVs identify the 8 targets in less than 150 seconds. Nevertheless, they are unable to determine whether all targets have been identified. To evaluate the number of targets N^t , the RoI has to be explored until $\bar{\mathbb{X}}_{i,k}^t = \emptyset$. As long as $\bar{\mathbb{X}}_{i,k}^t \neq \emptyset$, not yet identified target may still

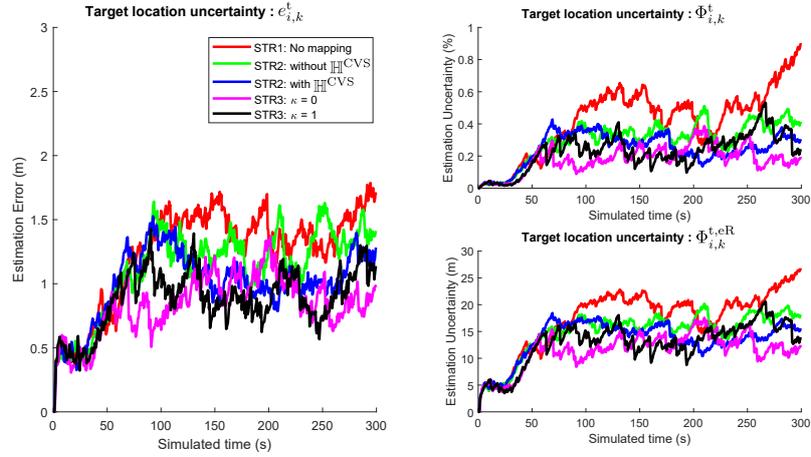


Figure 5.10 – Mean value over 10 simulations for each criterion of $e_{i,k}^t$ (left), $\Phi_{i,k}^t$ (top-right) and $\Phi_{i,k}^{t,eR}$ (bottom-right). To lighten the figure, the standard deviation is not shown.

criterion	(5.16)	(5.21) no \mathbb{H}^{CVS}	(5.21)	(5.27)	(5.32)
$e_{i,k}^t$ (m)	1.43 ± 0.59	1.28 ± 0.45	1.10 ± 0.40	0.92 ± 0.46	0.95 ± 0.23
$\Phi_{i,k}^{t,eR}$ (m)	18.29 ± 8.10	16.06 ± 4.48	13.57 ± 5.50	11.71 ± 4.63	13.77 ± 3.76

Table 5.3 – Mean value and standard deviation of $e_{i,k}^t$ and $\Phi_{i,k}^{t,eR}$ between $t = 150$ s and $t = 300$ s

belong to $\bar{\mathbb{X}}_{i,k}^t$.

Figure 5.11 shows the evolution of $\phi(\bar{\mathbb{X}}_{i,k}^t) / \phi(\mathbb{X}_g)$ (solid lines) for each criterion. Table 5.4 gathers the average of those quantities between $t = 150$ s and $t = 300$ s for each criterion. Even if $\phi(\bar{\mathbb{X}}_{i,k}^t)$ decreases quickly during the first time steps, no strategy is able to get $\phi(\bar{\mathbb{X}}_{i,k}^t) = 0$ so as to ensure that all targets have been found. Instead, this area still to explore seems to converge to a value related to the ability of the fleet to spread themselves and explore the unknown urban environment.

Figure 5.11 also shows the evolution of $\phi(\bar{\mathbb{X}}_{i,k}^t \cap \mathbb{X}_{i,k}^h) / \phi(\mathbb{X}_g)$ (dashed lines), where $\bar{\mathbb{X}}_{i,k}^t \cap \mathbb{X}_{i,k}^h$ represents the portion of $\bar{\mathbb{X}}_{i,k}^t$ that has been occluded by an obstacle and has not yet been observed again. According to Table 5.4, when the trajectory update is performed using (5.16), almost two thirds of $\bar{\mathbb{X}}_{i,k}^t$ remain hidden by an obstacle. The parts of $\bar{\mathbb{X}}_{i,k}^t$ intersecting $\mathbb{X}_{i,k}^h$ grow until they are observed from a different point of view. In fact, $42 \pm 4\%$ of \mathbb{X}_g may still be a possible location of an unidentified target. This clearly shows a limitation of this strategy that does not account for any kind of mapping of the obstacles.

When the UAV trajectory is designed using either the criterion (5.21) or (5.32), thus when $\mathbb{X}_{i,k}^o$ or the OEM are used, then the exploration of $\bar{\mathbb{X}}_{i,k}^t$ is largely improved. According to the results, the criterion (5.32), which involves a strategy favoring the exploration of occluded areas, is proved to be the best approach to search for unidentified targets.

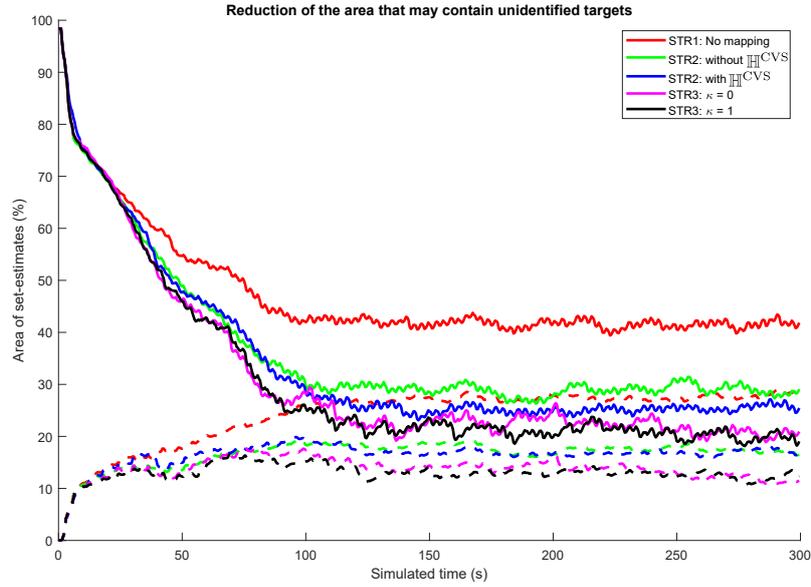


Figure 5.11 – Mean value over 10 simulations for each criterion of $\phi(\bar{\mathbb{X}}_{i,k}^t) / \phi(\mathbb{X}_g)$ (solid lines), $\phi(\bar{\mathbb{X}}_{i,k}^t \cap \mathbb{X}_{i,k}^h) / \phi(\mathbb{X}_g)$ (dashed lines). To lighten the figure, the standard deviation is not shown.

critereon	(5.16)	(5.21) no \mathbb{I}^{CVS}	(5.21)	(5.27)	(5.32)
$\phi(\bar{\mathbb{X}}_{i,k}^t) (\%)$	42.06 ± 4.21	29.56 ± 3.70	26.34 ± 2.88	23.14 ± 2.87	22.06 ± 2.35
$\phi(\bar{\mathbb{X}}_{i,k}^t \cap \mathbb{X}_{i,k}^h) (\%)$	26.88 ± 2.29	17.70 ± 2.49	17.20 ± 2.00	14.10 ± 1.78	13.18 ± 1.94

Table 5.4 – Mean value and standard deviation of $\phi(\bar{\mathbb{X}}_{i,k}^t) / \phi(\mathbb{X}_g)$ and $\phi(\bar{\mathbb{X}}_{i,k}^t \cap \mathbb{X}_{i,k}^h) / \phi(\mathbb{X}_g)$ between $t = 150$ s and $t = 300$ s

5.4.4 . Exploration efficiency

The UAV trajectory design strategies are now compared considering their performance regarding the exploration of the environment.

Figure 5.12 compares the evolution of $\Phi_k^{\text{FoV}} / \phi(\mathbb{X}_g)$ and of $\Phi_k^g / \phi(\mathbb{X}_g)$ while only considering criteria (5.16) (top) and (5.32) (bottom). Table 5.5 gathers the average of those metrics as well as the ratio

$$\bar{\Phi}^{\text{FoV}} = \frac{\text{mean}(\Phi_k^{\text{FoV}}) - \text{mean}(\Phi_k^g)}{\text{mean}(\Phi_k^{\text{FoV}})}$$

representing the averaged portion of the ground within the FoV of one UAV that is not monitored due to the presence of an obstacle. An efficient trajectory design strategy within a cluttered environment should be such that $\bar{\Phi}^{\text{FoV}}$ is small. This is because, the smaller $\bar{\Phi}^{\text{FoV}}$ is, the more information from the ground (observation of a target or the ground) is collected to evaluate the presence or the absence of a target.

Figure 5.12 and Table 5.5 illustrate the impact of the presence of obstacles within the RoI. Regarding criterion (5.16), the fleet theoretically monitors 30 % of the RoI. Nonetheless, due to the presence

of unknown obstacles, almost one third of this theoretically monitored area is actually occluded. Criterion (5.21) does not really improve the ratio $\bar{\Phi}^{\text{FoV}}$ since, unlike criterion (5.27), it cannot predict efficiently the part of the FoV that will be hidden by obstacles during the future time steps. The inclusion of $\mathbb{H}_{i,k}^{\text{CVS}}$ in (5.21) as the area that UAV i cannot observe during the next h time steps due to the presence of recently detected obstacles only improves slightly $\bar{\Phi}^{\text{FoV}}$. The criteria involving the OEM and, thus, the prediction of the hidden areas, proves to be the best approach to reduce the ratio $\bar{\Phi}^{\text{FoV}}$. The value of $\bar{\Phi}^{\text{FoV}}$ is larger when using criterion (5.27) than with criterion (5.32). This may be explained by the fact that since (5.32) favors trajectories leading to the exploration of areas near obstacles, the FoV of the UAVs controlled by this strategy will contain more often an obstacle.

The oscillations of the curves are due to the update of the trajectories with a period of 3 s.

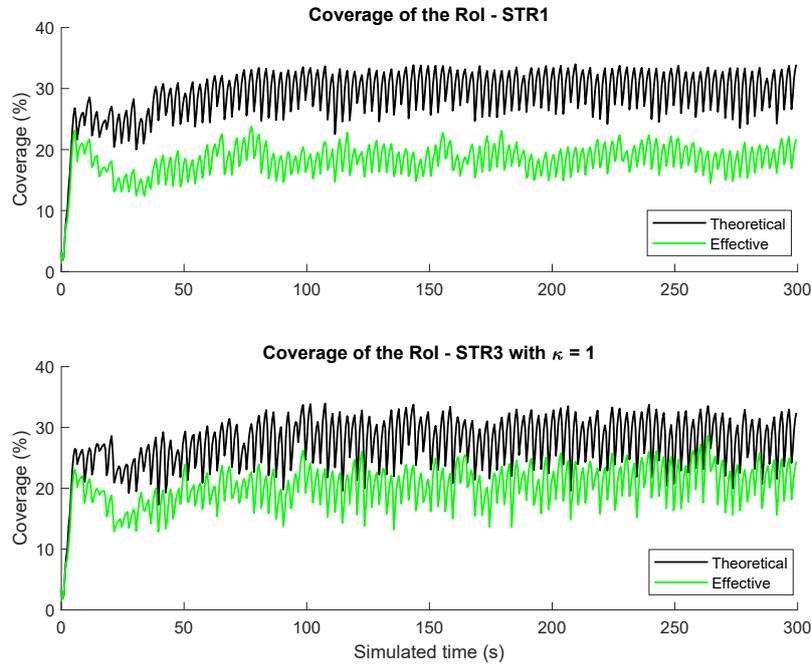


Figure 5.12 – Mean value of $\Phi_k^{\text{FoV}}/\phi(\mathbb{X}_g)$ (black) and of $\Phi_k^g/\phi(\mathbb{X}_g)$ (green) when only criteria (5.16) (top) and (5.32) (bottom) are considered

critérium	(5.16)	(5.21) no \mathbb{H}^{CVS}	(5.21)	(5.27)	(5.32)
Φ_k^{FoV} (%)	30.24 ± 2.24	30.13 ± 2.25	28.34 ± 2.00	28.66 ± 2.17	28.46 ± 2.12
Φ_k^g (%)	18.73 ± 3.27	19.38 ± 3.39	19.06 ± 3.35	21.63 ± 3.04	21.04 ± 3.02
$\bar{\Phi}^{\text{FoV}}$ (%)	38.06	35.68	32.73	24.55	26.07

Table 5.5 – Mean value and standard deviation of $\Phi_k^{\text{FoV}}/\phi(\mathbb{X}_g)$, $\Phi_k^g/\phi(\mathbb{X}_g)$ and $\bar{\Phi}^{\text{FoV}}$ between $t = 150$ s and $t = 300$ s

Figure 5.13 shows the evolution of the proportion of the \mathbb{X}_g that has been explored, *i.e.*, $\Phi_k^{\text{CG}}/\phi(\mathbb{X}_g)$. Table 5.6 gathers the average and standard deviation of this metric.

Remind that no controller presented in this chapter has been designed to simply explore the RoI. Our main interest is to search and localize the targets, not to map the RoI. Consequently, $\Phi_k^{Cg}/\phi(\mathbb{X}_g)$ is more a metric representing the capacity of the UAVs to efficiently evolve within the environment and explore the RoI than to evaluate the performance of these strategies to map the RoI.

As illustrated in Figure 5.13, $\Phi_k^{Cg}/\phi(\mathbb{X}_g)$ increases quickly at the beginning but, then, one can remark 3 distinct tendencies. The criterion (5.16) has the slowest increase since the strategy does not account for any kind of mapping. Criterion (5.21) leads to a faster exploration because the UAVs may avoid the obstacles since they are not attracted toward the area $\mathbb{X}_{i,k}^o$ around detected obstacles where no target can be. Nevertheless, and as expected, criterion (5.27) leads to the fastest increase of $\Phi_k^{Cg}/\phi(\mathbb{X}_g)$ since the UAVs are able to predict the portion of the ground that they will not see using the OEM, and thus efficiently move around obstacles. Moreover, one can have the same remark as for Figure 5.12, *i.e.*, the criterion (5.27) shows a slightly better performance regarding the metric $\Phi_k^{Cg}/\phi(\mathbb{X}_g)$ than criterion (5.32). This is because UAVs that update their trajectory by using criterion (5.32) are likely to stay around their neighbors in order to explore the area they did not observe due to the presence of an obstacle. Consequently, they explore less.

In addition, since 5% of \mathbb{X}_g is occupied by obstacles, the maximum theoretical value of $\Phi_k^{Cg}/\phi(\mathbb{X}_g)$ is 95%. This value is almost reached by the criteria at time $t = 300$ seconds, with 92.7 % with (5.16) to 93.5 % with criterion (5.27).

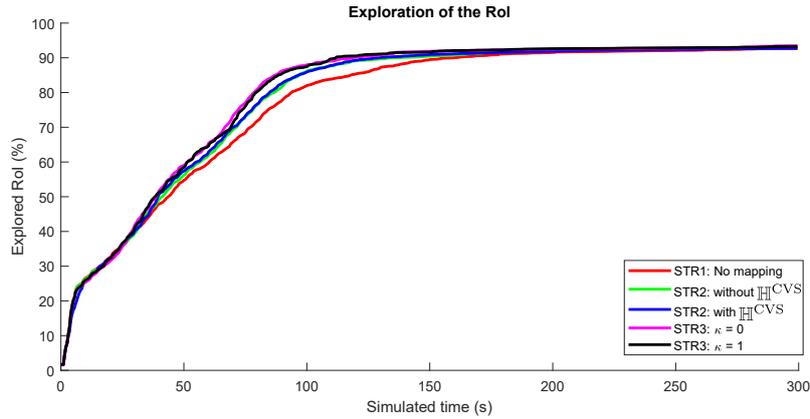


Figure 5.13 – Mean value over 10 simulations for each criterion of $\Phi_{i,k}^{Cg}/\phi(\mathbb{X}_g)$. To lighten the figure, the standard deviation is not shown.

critereon	(5.16)	(5.21) no \mathbb{H}^{CVS}	(5.21)	(5.27)	(5.32)
Φ_k^{Cg}	88.64 ± 2.01	90.14 ± 1.19	90.12 ± 1.56	91.15 ± 1.03	91.14 ± 1.19

Table 5.6 – Mean value and standard deviation of $\Phi_{i,k}^{Cg}/\phi(\mathbb{X}_g)$ between $t = 150$ s and $t = 300$ s

Figure 5.14 shows the evolution of Φ_k^{OEM} . At the beginning of the target search, Φ_k^{OEM} increases quickly but becomes more or less stable around $t = 100$ seconds. This is consistent with Figure 5.13 since most of the RoI has been explored and, thus, most of the obstacles have been detected. The mean and standard deviation of this metric when averaged between $t = 150$ s and $t = 300$ s is

74.38 ± 9.25 %. This means that around three quarters of the part of the ground that is predicted to be hidden by obstacles $\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_{i,k}^u) \cap \mathbb{F}(\mathbf{x}_{i,k}^u)$ is actually hidden when considering CVS information $\mathbb{H}_{i,k}^{\text{CVS}}$. It is likely that with an OEM of higher resolution, *i.e.*, with a cell dimension smaller than 5 m, the prediction would be of higher quality. Nonetheless, this result shows that a 2.5 D mapping approach can be used to efficiently predict which portion of the ground the UAV may or may not observe in the future.

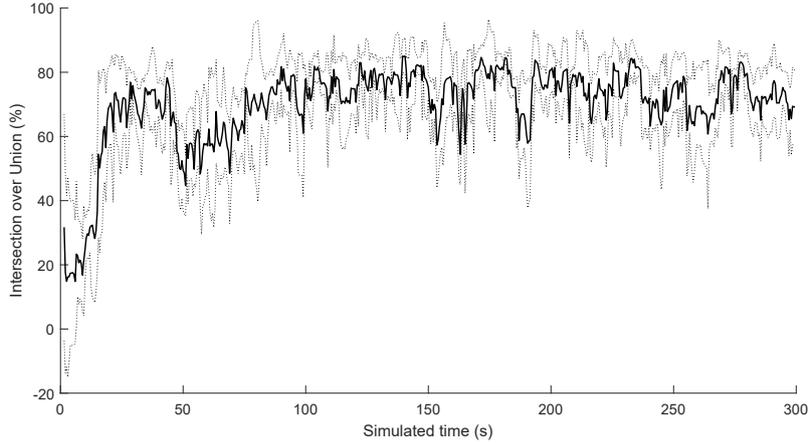


Figure 5.14 – Mean value and standard deviation over 10 simulations of Φ_k^{OEM} when only criterion (5.32) is considered

5.4.5 . Computing time

Figure 5.15 (left) shows the evolution of the average computing time required by a single UAV to process the CVS measurements and to update its knowledge. The time includes the characterization of the set estimates from the CVS measurements and the update of $\mathcal{X}_{i,k}^t$, $\bar{\Sigma}_{i,k}^t$, and $\Sigma_{i,k}^o$ (communication with neighbors included). We recall that we consider the sequential scheduling, see Figure 5.6, with a loopback to organize the computations and communications between UAVs. Thus, during the trajectory update, each UAV has to wait a certain amount of time after the first iteration of the MPC to receive the second updated trajectory of its neighbors. Figure 5.15 (right) does not account for this delay and only shows the average time required by a single UAV for the last iteration of the MPC. These results have been obtained on a computer embedding an Intel Core i7-10875H 2.30GHz. Consequently, they provide only a first idea of the time that would be actually required considering an implementation on a UAV.

The time that is required to process the CVS information is almost the same for criteria (5.21) (SRT2) and (5.32) (STR3), *i.e.*, about 2.15 seconds, and is almost three times that required for criterion (5.16) (STR1), *i.e.*, about 0.81 second, which does not exploit CVS information related to obstacles and does not characterize $\Sigma_{i,k}^o$ and $\Sigma_{i,k}^o$. The update of the OEM (the main difference between criteria (5.21) and (5.32)) does not take much time.

The time required by the MPC is almost the same for criteria (5.16) and (5.21), *i.e.*, about 0.90 seconds, but is around 1.52 seconds for criterion (5.32) (regardless of the value of κ). This is mainly due to the evaluation of $\mathbb{H}_{i,k}^{\text{OEM}}(\mathbf{x}_{i,k+\tau|k}^u)$, for $\tau \in \{1, \dots, h\}$, from the OEM in criterion (5.32). Consequently,

the improved performance in target search and tracking when we add this 2.5D mapping approach is counterbalanced by the additional time required to update the trajectories of the UAVs.

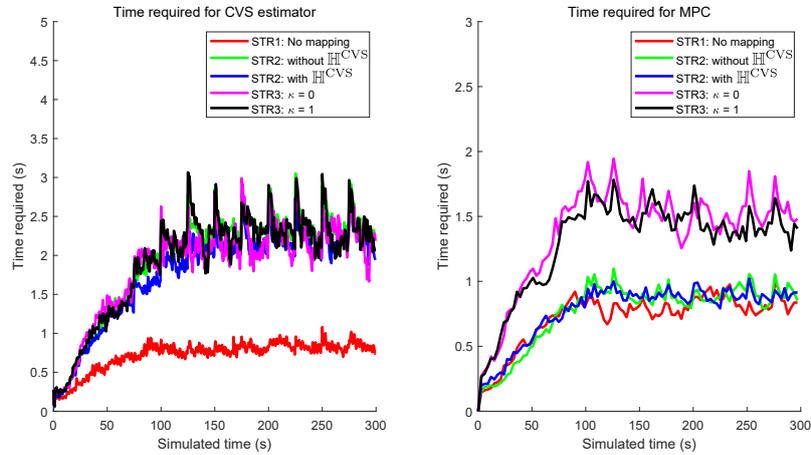


Figure 5.15 – Mean value over 10 simulations of the time required for a UAV to run the set estimator from CVS measurements (left) and the MPC approach (right)

5.5 . Conclusion

This chapter presents several strategies to update the trajectories of a fleet of UAVs in order to address a CSAT problem within an unknown environment. The targets are assumed to outnumber the UAVs. Therefore, a trade-off between searching for new targets and pursuing previously identified targets must be found. For that purpose, the chosen control strategy, based on a distributed MPC, aims to search for a sequence of control inputs that minimizes a criterion translating the main objective to achieve : the reduction of uncertainty regarding the estimated location of both identified and unidentified targets. The performance of the strategy proposed in (Ibenthal, 2022), represented by criterion (5.16), is evaluated. Nonetheless, this criterion does not take into account any available knowledge regarding the environment. Consequently, other criteria based on the one proposed in (Ibenthal, 2022) have been developed to account for different levels of available knowledge about the environment that the UAV may gain, namely

- No environmental knowledge : the UAVs are unable to gain any knowledge regarding the environment
- No memory regarding the shape of the obstacles. The UAVs are able to get some knowledge regarding the obstacles that are under observation, but no memory is allocated to concatenate this knowledge over time
- Mapping of a 3D environment. The UAVs embed the required devices, algorithms, and memory to update a map of the environment that approximates the shape of the obstacles.

The performance of the criteria that are derived considering these hypotheses have been compared. The more environmental knowledge is available, the better the strategy that exploits this knowledge is. The reason is during a target search within an unknown and cluttered RoI, a UAV has to design

a trajectory in order to move around obstacles to explore areas that may contain targets. Thus, the better the UAV knows about the location and shape of an obstacle, the better it can move around that obstacle. The simulations performed for this chapter are clear about this statement, even though the exploitation of this knowledge related to the obstacles induces a higher computational time for both the target localization and the trajectory update. Besides, the simulations have been performed in a idealized situation where the communications between the UAVs are without error and delay. We considered this idealized situation to evaluate the performance of our approach designed to solve a CSAT problem before including communication issues or other real-life problems which will undoubtedly reduce its target-search-and-track performance.

The main contribution of this chapter is the exploitation of the OEM, in criterion (5.27), to improve the target search within an unknown and cluttered environment. The OEM approximates the previously detected obstacles with boxes that are then used by the UAVs to better predict what they will observe in the future. According to the simulations, criterion (5.27) always provides better target-search-and-tracking performance than criteria (5.16) or (5.21). Nevertheless, due to the presence of obstacles, the UAVs cannot observe some portions of the RoI that may contain unidentified targets. To efficiently explore those unchecked regions, a UAV must circle around an obstacle faster than the targets. To address this path-planning problem, we added in criterion (5.27) a new term, *i.e.*, J_i^2 introduced in (5.31), to favor trajectories leading to the cooperative exploration of these recently hidden areas. Even though this new criterion, *i.e.*, (5.32), has shown promising results, the target-search-and-tracking performance is not fully satisfying. On the one hand, the criterion allows a better reduction of the hidden regions that may contain unidentified targets. On the other hand, the fact that the UAVs favor trajectories that lead to the exploration of recently hidden regions disadvantages trajectories that would allow a UAV to recapture a previously identified target in order to reduce its estimated location uncertainty.

Chapter 6 - Target localization with UAV state uncertainties

The approaches presented in Chapter 4 and Chapter 5 to search and localize ground targets assume that each UAV has an accurate knowledge of its state vector. Nevertheless, due to uncertainty in the dynamics and noise on the measurements provided by their embedded proprioceptive sensors, the UAVs only have an uncertain estimate of their state, see, e.g. (Vanegas et al., 2019; Zhu et al., 2021), and (Kenmogne et al., 2019).

In this chapter, it is assumed that each UAV disposes only of an estimation of its state vector and the uncertainty associated with this estimate is assumed bounded. As the CVS measurements used for target localization are initially expressed in a frame attached to each UAV and the set estimates containing the locations of the targets are expressed in a common reference frame, it is necessary to account for the UAV state uncertainties in the target localization problem. Consequently, the approaches introduced in Chapter 4 must be adapted to account for the UAV state uncertainty.

The set-membership estimator presented in Chapter 4 exploits the available CVS measurements to characterize set estimates that either are guaranteed to contain the location of a target $\mathbb{X}_{i,j,k}^{t,m}$ for each identified target $j \in \mathcal{D}_{i,k}^t$ at time t_k , or are guaranteed not to contain a target location at time t_k , e.g., $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$. These set estimates used for target localization are built considering the following sets

- $\mathbb{P}_{i,k}((n_r, n_c))$, see (6.4), used by UAV i to characterize $\mathbb{X}_{i,j,k}^{t,m}$ for each identified target $j \in \mathcal{D}_{i,k}^t$ as well as $\mathbb{X}_{i,k}^o$ and $\mathbb{X}_{i,j,k}^{t,m}$
- $\mathbb{P}_{i,k}^g((n_r, n_c))$, see (6.13), used by UAV i to characterize $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$ and $\mathbb{H}_{i,k}^{CVS}$.

The definitions of $\mathbb{P}_{i,k}((n_r, n_c))$ and $\mathbb{P}_{i,k}^g((n_r, n_c))$ and the methods used for their characterization need to be adapted to account for the UAV state uncertainty.

These problems are formalized in Section 6.1. Section 6.2 describes briefly the approach to solve this problem. Then, Sections 6.3, 6.4, and 6.5 propose a method to characterize $\mathbb{P}_{i,k}((n_r, n_c))$ and $\mathbb{P}_{i,k}^g((n_r, n_c))$ while being robust to UAV state uncertainty. These adapted set estimates are then used to make the target location estimator robust to UAV state uncertainty. Finally, Section 6.6 presents simulation results to evaluate the performance of the robust estimator.

6.1 . Problem formulation

As in the previous chapters, we consider that a fleet of N^u cooperative UAVs is tasked to find and localize N^t unknown ground targets in a RoI \mathbb{X}_0 . For that purpose, each UAV estimates the possible locations of the target by using the set-membership estimator presented in Chapter 4, and evolves within \mathbb{X}_0 according to a cooperative target search and tracking strategy detailed in Chapter 5.

6.1.1 . UAV state uncertainty

In this chapter, the dynamics of UAV i , with $i \in \mathcal{N}^u$, is modeled as

$$\mathbf{x}_{i,k+1}^u = \mathbf{f}^u(\mathbf{x}_{i,k}^u, \mathbf{u}_{i,k}^u, \mathbf{w}_{i,k}^u) \quad (6.1)$$

where $\mathbf{w}_{i,k}^u$ represents the state perturbations assumed unknown but bounded in a known box $[\mathbf{w}^u] = [\underline{\mathbf{w}}^u, \overline{\mathbf{w}}^u]$. Consequently, the UAV state vector $\mathbf{x}_{i,k}^u$ is only known to belong to the set estimate $\mathbb{X}_{i,k}^u$. This set estimate can be obtained with set-membership estimation techniques by using the uncertain dynamics (6.1) and $\mathbb{X}_{i,k-1}^u$, *i.e.*, the possible values of the state vector of UAV i known at time t_{k-1}

$$\mathbb{X}_{i,k|k-1}^u = \mathbf{f}^u(\mathbb{X}_{i,k-1}^u, \mathbf{u}_{i,k}^u, [\mathbf{w}^u]). \quad (6.2)$$

This prediction is then corrected by using the measurement vector $\mathbf{y}_{i,k}^u$ provided by proprioceptive sensors like an IMU or gyroscope, see (3.14), *i.e.*,

$$\mathbb{X}_{i,k|k}^u = \left\{ \mathbf{x}^u \in \mathbb{X}_{i,k|k-1}^u \mid \mathbf{h}^u(\mathbf{x}^u) \in (\{\mathbf{y}_{i,k}^u\} - [\mathbf{w}^{u,m}]) \right\}, \quad (6.3)$$

where \mathbf{h}^u is the measurement function associated to the proprioceptive sensors and $[\mathbf{w}^{u,m}]$ is some known box assumed to contain the measurement noise of those sensors. UAV i can improve the accuracy of the estimation of its state vector by accounting for information provided by neighboring UAVs considering a cooperative navigation, see (Vetrella et al., 2016). In what follows, we assume that, at time t_k , $\mathbb{X}_{i,k}^u$ is available and we focus on the impact of the UAV state uncertainty on the target localization accuracy.

6.1.2 . Exploiting CVS measurements

To lighten the notations, the time index is omitted.

The CVS measurements available for a pixel $(n_r, n_c) \in \mathcal{N}^1$ at time t_k consist of an interval $[\mathbf{D}_i](n_r, n_c)$ containing the depth information and a pixel label, *i.e.*, $(n_r, n_c) \in \mathcal{Y}_i^\ell$ with $\ell \in \{t, o, g, n\}$. A bounding box $[\mathcal{Y}_{i,j}^t]$ containing (n_r, n_c) may also be available when a target is identified. In Chapter 4, two sets have been introduced depending on the label of (n_r, n_c) . If $(n_r, n_c) \in [\mathcal{Y}_{i,j}^t] \cap \mathcal{Y}_i^t$ or $(n_r, n_c) \in \mathcal{Y}_i^o$, one may evaluate $\mathbb{P}_i((n_r, n_c))$, which has a non-empty intersection with a target or with an obstacle. If $(n_r, n_c) \in \mathcal{Y}_i^g$, one may evaluate $\mathbb{P}_i^g((n_r, n_c))$ to get a part of the ground free of target. Even though $\mathbb{H}_i^{\text{CVS}}$ is not used to localize the target, it is also obtained from $\mathbb{P}_i^g((n_r, n_c))$ when $(n_r, n_c) \in \mathcal{Y}_i^o \cup \mathcal{Y}_i^t \cup \mathcal{Y}_i^n$.

Our aim in what follows is to define $\mathbb{P}_i((n_r, n_c))$ and $\mathbb{P}_i^g((n_r, n_c))$ as the solution of set inversion problems that involves the uncertain UAV state.

Consider first the case of a pixel $(n_r, n_c) \in [\mathcal{Y}_{i,j}^t] \cap \mathcal{Y}_i^t$ or $(n_r, n_c) \in \mathcal{Y}_i^o$. For a given value \mathbf{x}^u of the state vector of UAV i , $\mathbb{P}_i((n_r, n_c))$ has been defined in Section 4.2 as

$$\mathbb{P}_i((n_r, n_c), \mathbf{x}^u) = \left\{ \mathbf{x} \in \mathbb{X}_0 \mid \mathbf{x} \in \mathbb{F}(\mathbf{x}^u), \exists \mathbf{v} \in \mathcal{V}_i(n_r, n_c), \right. \\ \left. d_v(\mathbf{x}_i^c, \{\mathbf{x}\}) \in [\mathbf{D}_i](n_r, n_c) \right\}, \quad (6.4)$$

where \mathbf{x}_i^c is the location of the optical center of the camera embedded on UAV i expressed in the reference frame \mathcal{F} . Moreover, since $\mathbf{x} \in \mathbb{F}(\mathbf{x}^u)$, then $\mathbf{x} \in \mathbb{B}(\mathbf{x}_i^c, d_{\max})$ according to (3.24), where $\mathbb{B}(\mathbf{x}_i^c, d_{\max})$ is a ball of center \mathbf{x}_i^c and radius d_{\max} . Consequently, according to (3.29), (6.4) can be re-defined as

$$\mathbb{P}_i((n_r, n_c), \mathbf{x}^u) = \left\{ \mathbf{x} \in \mathbb{X}_0 \mid \mathbf{p}_c(\mathbf{x}^u, \mathbf{x}) \in \begin{pmatrix} [n_c - 1, n_c] \\ [n_r - 1, n_r] \end{pmatrix}, \right. \\ \left. \|\mathbf{x}_i^c - \mathbf{x}\| \in [\mathbf{D}_i](n_r, n_c) \cap [0, d_{\max}] \right\}. \quad (6.5)$$

One can remark that $\|\mathbf{x}_i^c - \mathbf{x}\| = \left\| \mathbf{T}_{\mathcal{F}}^{\mathcal{F}^c}(\mathbf{x}) \right\|$, where $\mathbf{T}_{\mathcal{F}}^{\mathcal{F}^c}(\mathbf{x})$ is the transform (3.26) introduced in Chapter 3 that expresses in \mathcal{F}^c the coordinates of \mathbf{x} , initially expressed in \mathcal{F} . To get a more compact notation, we introduce the function

$$\mathbf{h}^{\text{CVS}}(\mathbf{x}, \mathbf{x}^u) = \begin{pmatrix} \left\| \mathbf{T}_{\mathcal{F}}^{\mathcal{F}^c}(\mathbf{x}) \right\| \\ \mathbf{p}_c(\mathbf{x}^u, \mathbf{x}) \end{pmatrix}, \quad (6.6)$$

and the box

$$[\mathbf{y}_{i,r,c}^{t/o}] = \begin{pmatrix} [\mathbf{D}_i](n_r, n_c) \cap [0, d_{\max}] \\ [n_c - 1, n_c] \\ [n_r - 1, n_r] \end{pmatrix}. \quad (6.7)$$

Thus, we have

$$\mathbb{P}_i((n_r, n_c), \mathbf{x}^u) = \{ \mathbf{x} \in \mathbb{X}_0 \mid \mathbf{h}^{\text{CVS}}(\mathbf{x}, \mathbf{x}^u) \in [\mathbf{y}_{i,r,c}^{t/o}] \}, \quad (6.8)$$

In 6.5, the dependency of \mathbb{P}_i in \mathbf{x}_i^u has been made explicit. Evaluating $\mathbb{P}_i((n_r, n_c), \mathbf{x}^u)$ requires the solution of a set inversion problem as $\mathbb{P}_i((n_r, n_c), \mathbf{x}^u)$ is the intersection of \mathbb{X}_0 with the preimage of the box $[\mathbf{y}_{i,r,c}^{t/o}]$ by the function \mathbf{h}^{CVS} , considered as a function of \mathbf{x} . Now, if the state of UAV i is only known to belong to some set \mathbb{X}_i^u , one may introduce the set of locations (in the RoI) that have been observed for *at least one* value $\mathbf{x}^u \in \mathbb{X}_i^u$ while being consistent with the depth measurement,

$$\mathbb{P}_i^{\exists}((n_r, n_c)) = \{ \mathbf{x} \in \mathbb{X}_0 \mid \exists \mathbf{x}^u \in \mathbb{X}_i^u, \mathbf{h}^{\text{CVS}}(\mathbf{x}, \mathbf{x}^u) \in [\mathbf{y}_{i,r,c}^{t/o}] \}, \quad (6.9)$$

$$= \bigcup_{\mathbf{x}^u \in \mathbb{X}_i^u} \mathbb{P}_i((n_r, n_c), \mathbf{x}^u), \quad (6.10)$$

and the set of locations that have been observed for *any* value of the UAV state $\mathbf{x}^u \in \mathbb{X}_i^u$ while being consistent with the depth measurement,

$$\mathbb{P}_i^{\forall}((n_r, n_c)) = \{ \mathbf{x} \in \mathbb{X}_0 \mid \forall \mathbf{x}^u \in \mathbb{X}_i^u, \mathbf{h}^{\text{CVS}}(\mathbf{x}, \mathbf{x}^u) \in [\mathbf{y}_{i,r,c}^{t/o}] \}, \quad (6.11)$$

$$= \bigcap_{\mathbf{x}^u \in \mathbb{X}_i^u} \mathbb{P}_i((n_r, n_c), \mathbf{x}^u). \quad (6.12)$$

These two sets can be interpreted as variants of $\mathbb{P}_i((n_r, n_c), \mathbf{x}^u)$ robust to the UAV state uncertainty.

Similarly, for the case of a pixel $(n_r, n_c) \in \mathcal{Y}_i^g$, and for a given value \mathbf{x}^u of the state vector of UAV i , $\mathbb{P}_i^g((n_r, n_c))$ has been defined in Section 4.3.1 as

$$\mathbb{P}_i^g((n_r, n_c), \mathbf{x}^u) = \left\{ \mathbf{x} \in \mathbb{X}_g \mid \mathbf{x} \in \mathbb{F}(\mathbf{x}^u), \mathbf{p}_c(\mathbf{x}^u, \mathbf{x}) \in \begin{pmatrix} [n_c - 1, n_c] \\ [n_r - 1, n_r] \end{pmatrix} \right\}. \quad (6.13)$$

Thus, (6.13), as with (6.4), can be redefined as

$$\mathbb{P}_i^g((n_r, n_c), \mathbf{x}^u) = \left\{ \mathbf{x} \in \mathbb{X}_0 \mid \mathbf{p}_c(\mathbf{x}^u, \mathbf{x}) \in \begin{pmatrix} [n_c - 1, n_c] \\ [n_r - 1, n_r] \end{pmatrix}, \|\mathbf{x}_i^c - \mathbf{x}\| \in [0, d_{\max}] \right\}, \quad (6.14)$$

$$= \left\{ \mathbf{x} \in \mathbb{X}_g \mid \mathbf{h}^{\text{CVS}}(\mathbf{x}, \mathbf{x}^u) \in [\mathbf{y}_{i,r,c}^g] \right\}, \quad (6.15)$$

with

$$\left[\mathbf{y}_{i,r,c}^g \right] = \begin{pmatrix} [0, d_{\max}] \\ [n_c - 1, n_c] \\ [n_r - 1, n_r] \end{pmatrix}. \quad (6.16)$$

In (6.13), the dependency of \mathbb{P}_i^g in \mathbf{x}^u has again been made explicit. Evaluating $\mathbb{P}_i^g((n_r, n_c), \mathbf{x}^u)$ requires the solution of a set inversion problem as $\mathbb{P}_i^g((n_r, n_c), \mathbf{x}^u)$ is the intersection of \mathbb{X}_g with the preimage of the box $\left[\mathbf{y}_{i,r,c}^g \right]$ by the function \mathbf{h}^{CVS} , considered as a function of \mathbf{x} . Now, if the state of UAV i is only known to belong to some set \mathbb{X}_i^u , one introduces the set of ground locations that may have contributed to the illumination of pixel (n_r, n_c) for *at least one* value $\mathbf{x}^u \in \mathbb{X}_i^u$,

$$\mathbb{P}_i^{g,\exists}((n_r, n_c)) = \left\{ \mathbf{x} \in \mathbb{X}_g \mid \exists \mathbf{x}^u \in \mathbb{X}_i^u, \mathbf{h}^{\text{CVS}}(\mathbf{x}, \mathbf{x}^u) \in \left[\mathbf{y}_{i,r,c}^g \right] \right\}. \quad (6.17)$$

$$= \bigcup_{\mathbf{x}^u \in \mathbb{X}_i^u} \mathbb{P}_i^g((n_r, n_c), \mathbf{x}^u), \quad (6.18)$$

and the set of locations that may have contributed to the illumination of pixel (n_r, n_c) for *any* value of the UAV state $\mathbf{x}^u \in \mathbb{X}_i^u$,

$$\mathbb{P}_i^{g,\forall}((n_r, n_c)) = \left\{ \mathbf{x} \in \mathbb{X}_g \mid \forall \mathbf{x}^u \in \mathbb{X}_i^u, \mathbf{h}^{\text{CVS}}(\mathbf{x}, \mathbf{x}^u) \in \left[\mathbf{y}_{i,r,c}^g \right] \right\}. \quad (6.19)$$

$$= \bigcap_{\mathbf{x}^u \in \mathbb{X}_i^u} \mathbb{P}_i^g((n_r, n_c), \mathbf{x}^u). \quad (6.20)$$

Again, (6.17) and (6.19) can be interpreted as variants of 6.13 robust to the UAV state uncertainty.

All of the set estimates introduced in Chapter 4 that are used to estimate the location of the targets are built upon $\mathbb{P}_{i,k}((n_r, n_c), \mathbf{x}_i^u)$ or $\mathbb{P}_{i,k}^g((n_r, n_c), \mathbf{x}_i^u)$, where \mathbf{x}_i^u is the true value of the state vector of UAV i . Thus, by replacing $\mathbb{P}_i((n_r, n_c), \mathbf{x}_i^u)$ with $\mathbb{P}_i^{\exists}((n_r, n_c))$, then one can be sure to get a set estimate that intersects the detected target or the detected obstacles, *i.e.*, because $\mathbb{P}_i((n_r, n_c), \mathbf{x}_i^u) \subset \mathbb{P}_i^{\exists}((n_r, n_c))$ for any $\mathbf{x}_i^u \in \mathbb{X}_i^u$. Similarly, by replacing (6.14) with (6.19), then, for a pixel labeled *Ground*, one can be sure to get a set estimate that represents the portion of the ground that contributed to the illumination of pixel (n_r, n_c) , whatever the values of the state of the UAV. This is due to the fact that $\mathbb{P}_i^{g,\forall}((n_r, n_c)) \subset \mathbb{P}_{i,k}^g((n_r, n_c), \mathbf{x}_i^u)$ if $\mathbf{x}_i^u \in \mathbb{X}_i^u$.

The sets $\mathbb{P}_i^{\forall}((n_r, n_c))$ and $\mathbb{P}_i^{g,\exists}((n_r, n_c))$ may also be useful to characterize. Nevertheless, replacing $\mathbb{P}_i((n_r, n_c), \mathbf{x}_i^u)$ with $\mathbb{P}_i^{\exists}((n_r, n_c))$ and replacing $\mathbb{P}_{i,k}^g((n_r, n_c), \mathbf{x}_i^u)$ with $\mathbb{P}_i^{g,\forall}((n_r, n_c))$ will make the characterization of $\mathbb{X}_{i,j,k}^{\text{t},m}$, $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$, $\mathbb{X}_{i,k}^o$, and $\mathbb{X}_{i,j,k}^{\text{t},m}$ robust to the UAV state uncertainty.

Sections 6.3 and 6.4 present a method to characterize $\mathbb{P}_i^{g,\exists}((n_r, n_c))$, $\mathbb{P}_i^{g,\forall}((n_r, n_c))$, $\mathbb{P}_i^{\exists}((n_r, n_c))$, and $\mathbb{P}_i^{\forall}((n_r, n_c))$.

6.2 . Interval analysis

We want to characterize $\mathbb{P}_i((n_r, n_c), \mathbf{x}^u)$ and $\mathbb{P}_i^g((n_r, n_c), \mathbf{x}^u)$, for a pixel $(n_r, n_c) \in \mathcal{N}^l$ while taking into account the fact that the true value of state vector of UAV i \mathbf{x}_i^u is only known to belong to \mathbb{X}_i^u . Various approaches are available to evaluate set estimates, *e.g.*, using ellipsoids or zonotopes, (Lu

and Zhou, 2017; Tang et al., 2020). Most are limited to models with linear measurement functions \mathbf{h} (Merhy, 2019). The set inversion problems $\mathbb{P}_i^{\mathbf{g},\exists}((n_r, n_c))$, $\mathbb{P}_i^{\mathbf{g},\forall}((n_r, n_c))$, $\mathbb{P}_i^{\exists}((n_r, n_c))$, and $\mathbb{P}_i^{\forall}((n_r, n_c))$ involve nonlinear functions. Consequently, we choose to use tools from interval analysis (Jaulin et al., 2001) and the Set Inverter Via Interval Analysis (SIVIA) (Jaulin and Walter, 1993) to approximate these sets by unions of non-overlapping boxes.

This section briefly recalls the basis of interval analysis and the way it may be used to characterize the sets such as $\mathbb{P}_i^{\mathbf{g},\exists}((n_r, n_c))$ or $\mathbb{P}_i^{\mathbf{g},\forall}((n_r, n_c))$. More details may be found in Appendices A.3 and A.4.

An interval $[x] = [\underline{x}, \bar{x}]$ of real numbers is the subset of \mathbb{R} such that

$$[x] = \{x \in \mathbb{R} \mid \underline{x} \leq x \text{ and } x \leq \bar{x}\}, \quad (6.21)$$

where \underline{x} and \bar{x} are respectively the lower and upper bounds of the interval. The set \mathbb{IR} contains all the real intervals. A box, denoted $[x]$, is a Cartesian product of intervals.

6.2.1 . Inclusion functions and set inversion

Consider a function $\mathbf{h} : \mathbb{X}_0 \times \mathbb{R}^n \rightarrow \mathbb{R}^m$ taking as inputs a vector \mathbf{x} and a vector of parameters \mathbf{p} . Assuming \mathbf{x} and \mathbf{p} are only known to belong to the boxes $[x_0] \subset \mathbb{X}_0$ and $[\mathbf{p}] \subset \mathbb{R}^n$, we may want to characterize the set containing the image of \mathbf{h} over $[x_0] \times [\mathbf{p}]$, *i.e.*,

$$\mathbf{h}([x_0], [\mathbf{p}]) = \{\mathbf{h}(\mathbf{x}, \mathbf{p}) \mid \mathbf{x} \in [x_0], \mathbf{p} \in [\mathbf{p}]\}. \quad (6.22)$$

This set can be impossible to characterize accurately if \mathbf{h} is nonlinear. To approximate $\mathbf{h}([x_0], [\mathbf{p}])$, we may use an inclusion function $[\mathbf{h}]$ of \mathbf{h} defined by interval analysis. An inclusion function takes intervals (or boxes) as inputs and returns an interval (or a box) as its output such that

$$\forall [x] \subset \mathbb{X}_0, \forall [\mathbf{p}] \subset \mathbb{R}^n, \text{ one has } \mathbf{h}([x], [\mathbf{p}]) \subset [\mathbf{h}]([x], [\mathbf{p}]). \quad (6.23)$$

Consider now that we want to characterize the set

$$\mathbb{X} = \{\mathbf{x} \in [x_0] \mid \mathbf{h}(\mathbf{x}, \mathbf{p}) \in [\mathbf{y}]\}, \quad (6.24)$$

for some known vector of parameters \mathbf{p} . If an inclusion function of \mathbf{h} is available, SIVIA can be used to get an inner and an outer approximation of \mathbb{X} using two unions of non-overlapping boxes (subpavings), see the example (A.21) and Figure A.6 in Appendix A.3. Actually, SIVIA builds iteratively three list of boxes or subpavings : \mathcal{B}_{in} , \mathcal{B}_{out} , and $\mathcal{B}_{\text{border}}$. The method to get these subpavings is

- If for a box $[x] \subset [x_0]$, one has $[\mathbf{h}]([x], \mathbf{p}) \subset [\mathbf{y}]$, then $[x] \subset \mathbb{X}$ and $[x] \in \mathcal{B}_{\text{in}}$
- If for a box $[x] \subset [x_0]$, one has $[\mathbf{h}]([x], \mathbf{p}) \cap [\mathbf{y}] = \emptyset$, then $[x] \cap \mathbb{X} = \emptyset$ and $[x] \in \mathcal{B}_{\text{out}}$
- If none of these two conditions is satisfied
 - if the width of $[x]$ is larger than some parameter ε , $[x]$ is bisected and the tests are applied on the resulting subboxes $[x]_2$ and $[x]_2$
 - else $[x] \in \mathcal{B}_{\text{border}}$

When the vector of parameters is only known to belong to $[\mathbf{p}]$, one may want to characterize one of the two following sets

$$\mathbb{X}^{\forall} = \{\mathbf{x} \in [x_0] \mid \forall \mathbf{p} \in [\mathbf{p}], \mathbf{h}(\mathbf{x}, \mathbf{p}) \in [\mathbf{y}]\}, \quad (6.25)$$

$$\mathbb{X}^{\exists} = \{\mathbf{x} \in [x_0] \mid \exists \mathbf{p} \in [\mathbf{p}], \mathbf{h}(\mathbf{x}, \mathbf{p}) \in [\mathbf{y}]\}. \quad (6.26)$$

One has clearly $\mathbb{X}^\forall \subset \mathbb{X}^\exists$. In our case, the uncertain vector of parameters is the uncertain UAV state vector.

An approach (that is useful for in the next section) to reformulate these sets is to introduce the set of functions

$$\mathcal{H}([\mathbf{p}]) = \{\mathbf{f} \in \mathcal{F}(\mathbb{X}_0, \mathbb{R}^m) \mid \exists \mathbf{p} \in [\mathbf{p}], \forall \mathbf{x} \in \mathbb{X}_0, \mathbf{f}(\mathbf{x}) = \mathbf{h}(\mathbf{x}, \mathbf{p})\}, \quad (6.27)$$

where $\mathcal{F}(\mathbb{X}_0, \mathbb{R}^m)$ is the set of functions from \mathbb{X}_0 to \mathbb{R}^m . Then, one has

$$\mathbb{X}^\forall = \{\mathbf{x} \in [\mathbf{x}_0] \mid \forall \mathbf{f} \in \mathcal{H}([\mathbf{p}]), \mathbf{f}(\mathbf{x}) \in [\mathbf{y}]\}, \quad (6.28)$$

$$\mathbb{X}^\exists = \{\mathbf{x} \in [\mathbf{x}_0] \mid \exists \mathbf{f} \in \mathcal{H}([\mathbf{p}]), \mathbf{f}(\mathbf{x}) \in [\mathbf{y}]\}. \quad (6.29)$$

For a box $[\mathbf{x}] \subset [\mathbf{x}_0]$, and assuming $[\mathbf{h}]$ is available, if $[\mathbf{h}]([\mathbf{x}], [\mathbf{p}]) \subset [\mathbf{y}]$, then, according to (6.23), one has $\mathbf{h}([\mathbf{x}], [\mathbf{p}]) \subset [\mathbf{y}]$. Consequently, for all $\mathbf{x} \in [\mathbf{x}]$ and for all $\mathbf{p} \in [\mathbf{p}]$, $\mathbf{h}(\mathbf{x}, \mathbf{p}) \in [\mathbf{y}]$. Thus, $[\mathbf{x}] \subset \mathbb{X}^\forall$ and $[\mathbf{x}] \in \mathcal{B}_{\text{in}}$. Similarly, if $[\mathbf{h}]([\mathbf{x}], [\mathbf{p}]) \cap [\mathbf{y}] = \emptyset$, then for all $\mathbf{x} \in [\mathbf{x}]$ and for all $\mathbf{p} \in [\mathbf{p}]$, $\mathbf{h}(\mathbf{x}, \mathbf{p}) \cap [\mathbf{y}] = \emptyset$. Thus, $[\mathbf{x}] \cap \mathbb{X}^\exists = \emptyset$ and $[\mathbf{x}] \in \mathcal{B}_{\text{out}}$.

Consequently, \mathcal{B}_{in} is an inner-approximation of \mathbb{X}^\forall and, since \mathcal{B}_{out} has no intersection with \mathbb{X}^\exists , and $[\mathbf{x}_0] \setminus \mathcal{B}_{\text{out}} = \mathcal{B}_{\text{in}} \cup \mathcal{B}_{\text{border}}$, $\mathcal{B}_{\text{in}} \cup \mathcal{B}_{\text{border}}$ is an outer-approximation of \mathbb{X}^\exists .

Nevertheless, to approximate either \mathbb{X}^\forall or \mathbb{X}^\exists with SIVIA, the algorithm has to characterize \mathcal{B}_{in} , \mathcal{B}_{out} , and $\mathcal{B}_{\text{border}}$, but the characterization of $\mathcal{B}_{\text{border}}$ may lead to a heavy computational burden when the vector of parameters is uncertain, see Figure A.7 in Appendix A.3. The reason is that SIVIA is unable to evaluate whether a box $[\mathbf{x}]$ is such that $[\mathbf{x}] \subset \mathbb{X}^\exists \setminus \mathbb{X}^\forall$. Besides, if $[\mathbf{p}]$ is large, then it is likely that $\mathbb{X}^\exists \setminus \mathbb{X}^\forall$ grows. Moreover, since $[\mathbf{x}_0] \setminus (\mathcal{B}_{\text{out}} \cup \mathcal{B}_{\text{in}}) = \mathcal{B}_{\text{border}}$, then $\mathcal{B}_{\text{border}}$ is also an outer-approximation of $\mathbb{X}^\exists \setminus \mathbb{X}^\forall$. Consequently, the more uncertain $[\mathbf{p}]$ is, the larger the number of boxes with width ε there are in $\mathcal{B}_{\text{border}}$ and, thus, the computational complexity increases.

6.2.2 . Thick intervals

The authors of (Desrochers and Jaulin, 2017) propose an adaptation of SIVIA to address set inversion problems involving an uncertain vector of parameters. Their approach has led to the introduction of thick intervals. This section briefly recalls thick intervals, thick functions, and how they can be used to get an inner-approximation of \mathbb{X}^\forall or an outer-approximation of \mathbb{X}^\exists more efficiently than with SIVIA. More details may be found in Appendix A.4.

A thick interval $\llbracket x \rrbracket$ is a set of intervals. The lower and upper bounds of the thick interval $\llbracket x \rrbracket$ have to belong to an interval of lower bounds $[x^-]$ and an interval of upper bounds and $[x^+]$, *i.e.*,

$$\llbracket x \rrbracket = \{[x, \bar{x}] \in \mathbb{I}\mathbb{R} \mid x \in [x^-] \text{ and } \bar{x} \in [x^+]\}. \quad (6.30)$$

A thick box is the Cartesian product of thick intervals.

Consider a thick interval $\llbracket x \rrbracket$ and an interval $[x] = [\inf([x^-]), \max([x^+])]$, where $\inf([x^-])$ is the lower bound of $[x^-]$ and $\max([x^+])$ is the upper bound of $[x^+]$. Then for an interval $[a] \in [x]$, one as clearly $[a] \in [x]$. Nevertheless, for an interval $[a] \in [x]$, we may not have $[a] \in \llbracket x \rrbracket$ if the lower bound of $[a]$ is not in $[x^-]$ and the upper bound of $[a]$ is not in $[x^+]$. Figure 6.1 illustrates this property for thick boxes.

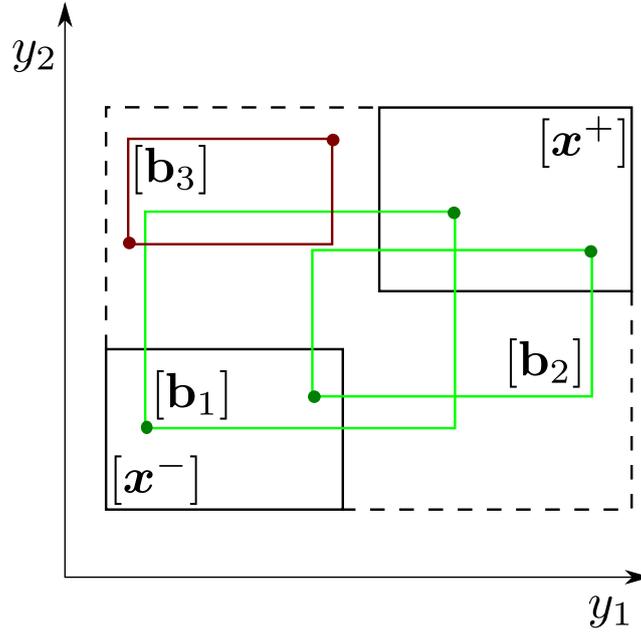


Figure 6.1 – Illustration of a thick box $\llbracket \mathbf{x} \rrbracket = \llbracket \mathbf{x}^-, \mathbf{x}^+ \rrbracket$ (not represented) and a box $[\mathbf{x}]$ (the dashed box). The boxes $[\mathbf{b}_1]$, $[\mathbf{b}_2]$, and $[\mathbf{b}_3]$ are included in $[\mathbf{x}]$, but $[\mathbf{b}_3]$ is not included in $\llbracket \mathbf{x} \rrbracket$ because its lower and upper bounds (represented by dots) are not in $[\mathbf{x}^-]$ and $[\mathbf{x}^+]$.

To get a thick box for a given function \mathbf{h} , one has to evaluate a thick function $\llbracket \mathbf{h} \rrbracket$ of \mathbf{h} . A thick function is a set of functions \mathbf{f} such that their images, for an input \mathbf{x} , always belong to a box $[\mathbf{h}^-(\mathbf{x}), \mathbf{h}^+(\mathbf{x})]$ of known bounds, *i.e.*,

$$\llbracket \mathbf{h} \rrbracket = \{ \mathbf{f} \in \mathcal{F}(\mathbb{X}_0, \mathbb{R}^m) \mid \forall \mathbf{x} \in \mathbb{X}_0, \mathbf{h}^-(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}) \leq \mathbf{h}^+(\mathbf{x}) \}. \quad (6.31)$$

Then, by evaluating the inclusion functions of \mathbf{h}^- and \mathbf{h}^+ , one gets a Thick Inclusion Function (TIF) $\llbracket \mathbf{h} \rrbracket$ of \mathbf{h} . The image of some $[\mathbf{x}] \subset \mathbb{X}_0$ by the TIF $\llbracket \mathbf{h} \rrbracket$ is a thick box, *i.e.*, set of boxes $[\underline{\mathbf{y}}, \bar{\mathbf{y}}] \in \mathbb{I}\mathbb{R}^m$ such that $\underline{\mathbf{y}}$ belong to $[\mathbf{h}^-]([\mathbf{x}])$ and $\bar{\mathbf{y}}$ belongs to $[\mathbf{h}^+]([\mathbf{x}])$, *i.e.*,

$$\llbracket \mathbf{h} \rrbracket([\mathbf{x}]) = \{ [\underline{\mathbf{y}}, \bar{\mathbf{y}}] \in \mathbb{I}\mathbb{R}^m \mid \underline{\mathbf{y}} \in [\mathbf{h}^-]([\mathbf{x}]), \bar{\mathbf{y}} \in [\mathbf{h}^+]([\mathbf{x}]) \}. \quad (6.32)$$

Consequently, if one chooses \mathbf{h}^- and \mathbf{h}^+ such that $\mathcal{H}([\mathbf{p}]) \subset \llbracket \mathbf{h} \rrbracket$, then one can define the sets

$$\underline{\mathbb{X}}^\forall = \{ \mathbf{x} \in [\mathbf{x}_0] \mid \forall \mathbf{f} \in \llbracket \mathbf{h} \rrbracket, \mathbf{f}(\mathbf{x}) \in [\mathbf{y}] \}, \quad (6.33)$$

$$\bar{\mathbb{X}}^\exists = \{ \mathbf{x} \in [\mathbf{x}_0] \mid \exists \mathbf{f} \in \llbracket \mathbf{h} \rrbracket, \mathbf{f}(\mathbf{x}) \in [\mathbf{y}] \}. \quad (6.34)$$

One has $\underline{\mathbb{X}}^\exists \subset \bar{\mathbb{X}}^\exists$ and $\underline{\mathbb{X}}^\forall \subset \bar{\mathbb{X}}^\forall$.

The approach proposed in (Desrochers and Jaulin, 2017) to get more efficiently an inner approximation of $\underline{\mathbb{X}}^\forall$ and an outer-approximation of $\bar{\mathbb{X}}^\exists$ consists in using a TIF of \mathbf{h} to characterize an inner approximation of $\underline{\mathbb{X}}^\forall$ and an outer approximation of $\bar{\mathbb{X}}^\exists$. For that aim, the authors of (Desrochers and Jaulin, 2017) propose the Thick SIVIA algorithm that evaluates two subpavings \mathcal{B}_{in} and \mathcal{B}_{out} using the following tests that involve a TIF $\llbracket \mathbf{h} \rrbracket$ of \mathbf{h} :

- If for a box $[x] \subset [x_0]$, one has $\llbracket \mathbf{h} \rrbracket ([x]) \subset [y]$, then $[x] \subset \underline{\mathbb{X}}^\vee$ and $[x] \in \mathcal{B}_{\text{in}}$
- If for a box $[x] \subset [x_0]$, one has $\llbracket \mathbf{h} \rrbracket ([x]) \cap [y] = \emptyset$ then $[x] \cap \overline{\mathbb{X}}^\exists = \emptyset$ and $[x] \in \mathcal{B}_{\text{out}}$

Additionally, if for a box $[x] \subset [x_0]$, one has $\llbracket \mathbf{h} \rrbracket ([x]) \cap [y] \neq \emptyset$, then one can prove that $[x] \subset \overline{\mathbb{X}}^\exists$ (see Sections A.4.3 and A.4.4 in Appendix A.4 for more details for this test). Consequently, a new subpaving $\mathcal{B}_{\text{penumbra}}$ containing those boxes can be characterized. The method to get $\mathcal{B}_{\text{border}}$ is unchanged, *i.e.*, if a box $[x] \subset [x_0]$ does not satisfy the three aforementioned tests, then it is bisected and its resulting boxes are retested until they are considered too small.

In Appendix A.4, Section A.4.3 presents a version of Thick SIVIA optimized to characterize $\overline{\mathbb{X}}^\exists$. In that section, all boxes $[x] \in \mathcal{B}_{\text{penumbra}}$ are such that $[x] \subset \overline{\mathbb{X}}^\exists$. Section A.4.4 presents another version of Thick SIVIA to characterize $\overline{\mathbb{X}}^\exists$ and $\underline{\mathbb{X}}^\vee$. In that section, all boxes $[x] \in \mathcal{B}_{\text{penumbra}}$ are such that $[x] \subset \overline{\mathbb{X}}^\exists \setminus \underline{\mathbb{X}}^\vee$.

This third test, *i.e.*, the one to get $\mathcal{B}_{\text{penumbra}}$, has been made possible only because thick boxes are used, see Proposition 30 in Appendix A.4. In fact, for a thick box $\llbracket x \rrbracket$, this test relies on the upper bound of $[x^-]$ and the lower bound of $[x^+]$. Those bounds are not available with a classic interval. Moreover, it is this third test that solves the problem of SIVIA because all boxes $[x] \in \mathcal{B}_{\text{penumbra}}$ are such that $[x] \subset \overline{\mathbb{X}}^\exists$ (or $[x] \subset \overline{\mathbb{X}}^\exists \setminus \underline{\mathbb{X}}^\vee$ depending of the version of Thick SIVIA that is used). Consequently, the characterization of $\overline{\mathbb{X}}^\exists \setminus \underline{\mathbb{X}}^\vee$ is more efficient.

As illustrated in Figures A.9 and A.10, Thick SIVIA is able to provide more efficiently an outer approximation of $\overline{\mathbb{X}}^\exists$, and thus \mathbb{X}^\exists , as well as an inner-approximation of $\underline{\mathbb{X}}^\vee$, and thus \mathbb{X}^\vee , than SIVIA. We use Thick SIVIA in what follows to characterize $\mathbb{P}_i^\vee((n_r, n_c))$, $\mathbb{P}_i^\exists((n_r, n_c))$, $\mathbb{P}_i^{\exists\vee}((n_r, n_c))$, and $\mathbb{P}_i^{\exists}((n_r, n_c))$, for any pixel $(n_r, n_c) \in \mathcal{N}^1$ while taking the uncertain UAV state vector into account.

Besides, since we use interval analysis, we consider that \mathbb{X}_i^u can be defined or approximated by a box $[x_i^u]$ such that $\mathbb{X}_i^u \subseteq [x_i^u]$.

6.3 . Thick function of \mathbf{h}^{CVS}

The sets $\mathbb{P}_i^\vee((n_r, n_c))$, $\mathbb{P}_i^\exists((n_r, n_c))$, $\mathbb{P}_i^{\exists\vee}((n_r, n_c))$, and $\mathbb{P}_i^{\exists}((n_r, n_c))$ can be characterized using SIVIA, see Appendix A.3. Nevertheless, SIVIA is not efficient enough when the set to characterize involves an uncertain vector of parameters, here it is the uncertain UAV state. The Thick SIVIA algorithm has been developed to solve the problem of efficiency of SIVIA, but, instead of an inclusion function of \mathbf{h}^{CVS} , it requires a thick inclusion function of \mathbf{h}^{CVS} . This section presents a method to get this thick inclusion function.

According to Appendix A.4.2, a thick function $\llbracket \mathbf{h}^{\text{CVS}} \rrbracket$ for \mathbf{h}^{CVS} is obtained by characterizing two functions $\mathbf{h}^{\text{CVS},-}$ and $\mathbf{h}^{\text{CVS},+}$ such that

$$\forall \mathbf{x}^u \in [x_i^u], \forall \mathbf{x} \in \mathbb{X}_0, \mathbf{h}^{\text{CVS},-}(\mathbf{x}) \leq \mathbf{h}^{\text{CVS}}(\mathbf{x}, \mathbf{x}^u) \leq \mathbf{h}^{\text{CVS},+}(\mathbf{x}). \quad (6.35)$$

Then, to get a TIF $\llbracket \mathbf{h}^{\text{CVS}} \rrbracket$ of \mathbf{h}^{CVS} one must get the inclusion functions $\llbracket \mathbf{h}^{\text{CVS},+} \rrbracket$ and $\llbracket \mathbf{h}^{\text{CVS},-} \rrbracket$ of $\mathbf{h}^{\text{CVS},+}$ and $\mathbf{h}^{\text{CVS},-}$. This can be achieved by evaluating the natural inclusion functions of $\mathbf{h}_{r,c}^+$ and $\mathbf{h}_{r,c}^-$, *c.f.*, Appendix A.3.2.

This section only presents a method to obtain the upper bound $\mathbf{h}^{\text{CVS},+}$. The approach to obtain the lower-bound $\mathbf{h}^{\text{CVS},-}$ is similar.

Consider a $\mathbf{x} \in \mathbb{X}_0$ and a $\mathbf{x}^u \in [\mathbf{x}_i^u]$, one has

$$\mathbf{h}^{\text{CVS}}(\mathbf{x}, \mathbf{x}^u) = \begin{pmatrix} \|\mathbf{T}_{\mathcal{F}}^{\mathcal{F}_i^c}(\mathbf{x})\| \\ (\mathbf{p}_c(\mathbf{x}^u, \mathbf{x}))_1 \\ (\mathbf{p}_c(\mathbf{x}^u, \mathbf{x}))_2 \end{pmatrix}, \quad (6.36)$$

where $(\mathbf{p}_c(\mathbf{x}^u, \mathbf{x}))_\ell$ is the ℓ -th coordinate of $\mathbf{p}_c(\mathbf{x}^u, \mathbf{x})$. One can remark that, in the transform

$$\mathbf{T}_{\mathcal{F}}^{\mathcal{F}_i^c}(\mathbf{x}) = \mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}_i^c} \left(\mathbf{M}_{\mathcal{F}}^{\mathcal{F}_i^b}(\mathbf{x} - \mathbf{x}^u) - \mathbf{x}_i^{c, \mathcal{F}_i^b} \right), \quad (6.37)$$

$$= \mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}_i^c} \mathbf{M}_{\mathcal{F}}^{\mathcal{F}_i^b} \left(\mathbf{x} - \left(\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}} \mathbf{x}_i^{c, \mathcal{F}_i^b} + \mathbf{x}^u \right) \right), \quad (6.38)$$

the rotation matrix $\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}_i^c}$ and $\mathbf{x}_i^{c, \mathcal{F}_i^b}$ are assumed to be perfectly known. $\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}}$ depends on the uncertain attitude of UAV i and \mathbf{x}_i^u is its the uncertain location. Then, according to (3.27), one has $\mathbf{p}_c(\mathbf{x}_i^u, \mathbf{x}) = \mathbf{p}_{\mathcal{F}_i^c}(\mathbf{T}_{\mathcal{F}}^{\mathcal{F}_i^c}(\mathbf{x}))$, where

$$\mathbf{p}_{\mathcal{F}_i^c}(\mathbf{x}^{\mathcal{F}_i^c}) = \begin{pmatrix} -f_c & 0 & N_c/2 \\ 0 & -f_r & N_r/2 \end{pmatrix} \begin{pmatrix} x_1^{\mathcal{F}_i^c}/x_3^{\mathcal{F}_i^c} \\ x_2^{\mathcal{F}_i^c}/x_3^{\mathcal{F}_i^c} \\ 1 \end{pmatrix}, \quad (6.39)$$

is a function that returns the 2D coordinates of the projection of some point $\mathbf{x}^{\mathcal{F}_i^c} \in \mathbb{R}^3$ on the CCD array

Then, consider a $\mathbf{x} \in \mathbb{X}_0$

$$\mathbf{h}^{\text{CVS},+}(\mathbf{x}) = \max_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \mathbf{h}^{\text{CVS}}(\mathbf{x}, \mathbf{x}^u), \quad (6.40)$$

$$= \begin{pmatrix} \max_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \left\| \mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}_i^c} \mathbf{M}_{\mathcal{F}}^{\mathcal{F}_i^b} \left(\mathbf{x} - \left(\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}} \mathbf{x}_i^{c, \mathcal{F}_i^b} + \mathbf{x}^u \right) \right) \right\| \\ \max_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \left(-f_c \frac{x_1^{\mathcal{F}_i^c}}{x_3^{\mathcal{F}_i^c}} + N_c/2 \right) \\ \max_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \left(-f_r \frac{x_2^{\mathcal{F}_i^c}}{x_3^{\mathcal{F}_i^c}} + N_r/2 \right) \end{pmatrix}, \quad (6.41)$$

$$= \begin{pmatrix} \max_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \left\| \mathbf{x} - \left(\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}} \mathbf{x}_i^{c, \mathcal{F}_i^b} + \mathbf{x}^u \right) \right\| \\ -f_c \min_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \left(\frac{x_1^{\mathcal{F}_i^c}}{x_3^{\mathcal{F}_i^c}} \right) + N_c/2 \\ -f_r \min_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \left(\frac{x_2^{\mathcal{F}_i^c}}{x_3^{\mathcal{F}_i^c}} \right) + N_r/2 \end{pmatrix}. \quad (6.42)$$

The evaluation of the maximum or the minimum of the norm $\|\mathbf{T}_{\mathcal{F}}^{\mathcal{F}_i^c}(\mathbf{x})\|$ and the evaluation of the maximum or the minimum of the ratio $x_{\{1,2\}}^{\mathcal{F}_i^c}/x_3^{\mathcal{F}_i^c}$ over the $[\mathbf{x}_i^u]$ is difficult. Instead, we propose

in Section 6.3.1 and 6.3.2 an approach to obtain a lower and upper bound of these minimum and maximum and, thus, an upper bound of $\mathbf{h}^{\text{CVS},+}$ and an lower bound of $\mathbf{h}^{\text{CVS},-}$.

Once the upper bound of $\mathbf{h}^{\text{CVS},+}$ and the lower bound of $\mathbf{h}^{\text{CVS},-}$ are available, one can characterize the natural inclusion functions of the upper bound of $\mathbf{h}^{\text{CVS},+}$ and the lower bound of $\mathbf{h}^{\text{CVS},-}$ to deduce a TIF of \mathbf{h}^{CVS} .

6.3.1 . Upper bound for the maximum of a norm

Instead of evaluating $\max_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \left\| \mathbf{x} - \left(\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}} \mathbf{x}_{i,k}^{c,\mathcal{F}_i^b} + \mathbf{x}^u \right) \right\|$, we propose to evaluate an upper bound.

Proposition 20 is introduced to get this upper bound.

Proposition 20. *Consider a series of functions $g_i(p) : \mathbb{R} \rightarrow \mathbb{R}$, for $i \in \{1, \dots, m\}$. such that*

$$\max_p \sum_{i=1}^m g_i(p) \leq \sum_{i=1}^m \max_p g_i(p) \quad (6.43)$$

$$\min_p \sum_{i=1}^m g_i(p) \geq \sum_{i=1}^m \min_p g_i(p) \quad (6.44)$$

One has

$$\max_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \left\| \mathbf{x} - \left(\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}} \mathbf{x}_i^{c,\mathcal{F}_i^b} + \mathbf{x}^u \right) \right\| = \sqrt{\max_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \left(\sum_{\ell=1}^3 \left(x_\ell - x_\ell^u - \left(\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}} \mathbf{x}_i^{c,\mathcal{F}_i^b} \right)_\ell \right)^2 \right)}, \quad (6.45)$$

where $(\mathbf{M}\mathbf{x})_\ell$ is the ℓ -th coordinate of the vector that results from the matrix-vector product $\mathbf{M}\mathbf{x}$. Then, with Proposition 20, one has

$$\max_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \left\| \mathbf{x} - \left(\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}} \mathbf{x}_i^{c,\mathcal{F}_i^b} + \mathbf{x}^u \right) \right\| \leq \sqrt{\sum_{\ell=1}^3 \max_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \left(x_\ell - x_\ell^u - \left(\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}} \mathbf{x}_i^{c,\mathcal{F}_i^b} \right)_\ell \right)^2}. \quad (6.46)$$

Then, the maximum of this quadratic function can be computed using interval analysis.

6.3.2 . Upper bound for the maximum of a ratio

The maximum of the ratio $[y] / [x]$ between two intervals $[x] = [\underline{x}, \bar{x}]$ and $[y] = [\underline{y}, \bar{y}]$ lies in the set $\left\{ \frac{\underline{y}}{\underline{x}}, \frac{\underline{y}}{\bar{x}}, \frac{\bar{y}}{\underline{x}}, \frac{\bar{y}}{\bar{x}} \right\}$. Therefore, the evaluation of $\max_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \left(x_2^{\mathcal{F}_i^c} / x_1^{\mathcal{F}_i^c} \right)$ is equivalent to compute the extrema of each coordinates of $\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}_i^c} \left(\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}_i^c} (\mathbf{x} - \mathbf{x}_i^u) - \mathbf{x}^{c,\mathcal{F}_i^b} \right)$. For that purpose, we introduce the following proposition

Proposition 21. *Consider a function $f(p) : \mathbb{R} \rightarrow \mathbb{R}$ and a function $g(x) : \mathbb{R} \rightarrow \mathbb{R}$. We have for all $x \in \mathbb{R}$*

$$\max_p (g(x) f(p)) = \max \left\{ g(x) \max_p (f(p)), g(x) \min_p (f(p)) \right\} \quad (6.47)$$

$$\min_p (g(x) f(p)) = \min \left\{ g(x) \min_p (f(p)), g(x) \max_p (f(p)) \right\} \quad (6.48)$$

Démonstration. In (6.47), $\max_p (g(x) f(p))$ depends of the sign of $g(x)$. If $g(x) \geq 0$, then

$$\max_p (g(x) f(p)) = g(x) \max_p (f(p)). \quad (6.49)$$

Moreover, since $\min_p (f(p)) \leq \max_p (f(p))$, then $g(x) \min_p (f(p)) \leq g(x) \max_p (f(p))$. If $g(x) < 0$, then

$$\max_p (g(x) f(p)) = g(x) \min_p (f(p)). \quad (6.50)$$

Moreover, since $\min_p (f(p)) \leq \max_p (f(p))$, then $g(x) \max_p (f(p)) \leq g(x) \min_p (f(p))$. Consequently, we have for all $x \in \mathbb{R}$

$$\max_p (g(x) f(p)) = \max \left\{ g(x) \max_p (f(p)), g(x) \min_p (f(p)) \right\}. \quad (6.51)$$

The same approach is used to prove (6.48). \square

According to Proposition 20, one has

$$\begin{aligned} \max_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \left(\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}_i^c} \left(\mathbf{M}_{\mathcal{F}}^{\mathcal{F}_i^b} (\mathbf{x} - \mathbf{x}_i^u) - \mathbf{x}_i^{c, \mathcal{F}_i^b} \right) \right) &\leq \max_{\Theta \in [\Theta_i^u]} \left(\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}_i^c} \mathbf{M}_{\mathcal{F}}^{\mathcal{F}_i^b} \mathbf{x} \right) - \min_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \left(\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}_i^c} \mathbf{M}_{\mathcal{F}}^{\mathcal{F}_i^b} \mathbf{x}^u \right) \\ &\quad - \min_{\mathbf{x}^u \in [\mathbf{x}_i^u]} \left(\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}_i^c} \mathbf{x}_i^{c, \mathcal{F}_i^b} \right), \end{aligned} \quad (6.52)$$

where $[\Theta_i^u]$ is a box containing all the possible attitudes of UAV i . The right term is independent of the UAV state. Therefore, it is constant. The middle term can be obtained using interval analysis. For the left term, we first introduce the elements of the matrix $\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}_i^c} \mathbf{M}_{\mathcal{F}}^{\mathcal{F}_i^b}$ as follows

$$\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}_i^c} \mathbf{M}_{\mathcal{F}}^{\mathcal{F}_i^b} = \begin{pmatrix} e_{11}(\Theta) & e_{12}(\Theta) & e_{13}(\Theta) \\ e_{21}(\Theta) & e_{22}(\Theta) & e_{23}(\Theta) \\ e_{31}(\Theta) & e_{32}(\Theta) & e_{33}(\Theta) \end{pmatrix}. \quad (6.53)$$

Then, we have

$$\max_{\Theta \in [\Theta_i^u]} \left(\mathbf{M}_{\mathcal{F}_i^b}^{\mathcal{F}_i^c} \mathbf{M}_{\mathcal{F}}^{\mathcal{F}_i^b} \mathbf{x} \right) = \begin{cases} \max_{\Theta \in [\Theta_i^u]} (e_{11}(\Theta) x_1 + e_{12}(\Theta) x_2 + e_{13}(\Theta) x_3) \\ \max_{\Theta \in [\Theta_i^u]} (e_{21}(\Theta) x_1 + e_{22}(\Theta) x_2 + e_{23}(\Theta) x_3) \\ \max_{\Theta \in [\Theta_i^u]} (e_{31}(\Theta) x_1 + e_{32}(\Theta) x_2 + e_{33}(\Theta) x_3) \end{cases} \quad (6.54)$$

$$\leq \begin{cases} \max_{\Theta \in [\Theta_i^u]} (e_{11}(\Theta) x_1) + \max_{\Theta \in [\Theta_i^u]} (e_{12}(\Theta) x_2) + \max_{\Theta \in [\Theta_i^u]} (e_{13}(\Theta) x_3) \\ \max_{\Theta \in [\Theta_i^u]} (e_{21}(\Theta) x_1) + \max_{\Theta \in [\Theta_i^u]} (e_{22}(\Theta) x_2) + \max_{\Theta \in [\Theta_i^u]} (e_{23}(\Theta) x_3) \\ \max_{\Theta \in [\Theta_i^u]} (e_{31}(\Theta) x_1) + \max_{\Theta \in [\Theta_i^u]} (e_{32}(\Theta) x_2) + \max_{\Theta \in [\Theta_i^u]} (e_{33}(\Theta) x_3) \end{cases} \quad (6.55)$$

Finally, Proposition 21 can be used to get an upper bound of this maximum.

6.4 . Characterization of $\mathbb{P}_i((n_r, n_c))$ and $\mathbb{P}_i^{\mathcal{G}}((n_r, n_c))$ accounting for UAV state uncertainties

With a TIF of \mathbf{h}^{CVS} available, one can use Thick SIVIA to efficiently characterize $\mathbb{P}_i^{\mathcal{G}, \exists}((n_r, n_c))$, $\mathbb{P}_i^{\exists}((n_r, n_c))$, $\mathbb{P}_i^{\mathcal{G}, \forall}((n_r, n_c))$, and $\mathbb{P}_i^{\forall}((n_r, n_c))$.

This section explains which of these sets can be used to replace $\mathbb{P}_i((n_r, n_c), \mathbf{x}_i^u)$ and $\mathbb{P}_i^{\mathcal{G}}((n_r, n_c), \mathbf{x}_i^u)$, with \mathbf{x}_i^u being the true value of UAV i state vector, to make the target location estimator robust to UAV state uncertainties.

6.4.1 . Characterization of $\mathbb{P}_i((n_r, n_c))$

Assuming the label of the pixel (n_r, n_c) is either *Target* or *Obstacle*, then $\mathbb{P}_i((n_r, n_c), \mathbf{x}^u)$ must intersect the shape of the detected target or the detected obstacle regardless of the UAV state $\mathbf{x}^u \in [\mathbf{x}_i^u]$. According to (6.9), one has $\mathbb{P}_i((n_r, n_c), \mathbf{x}^u) \subset \mathbb{P}_i^{\exists}((n_r, n_c))$ for all $\mathbf{x}^u \in [\mathbf{x}_i^u]$. Consequently, if $\mathbf{x}_i^u \in [\mathbf{x}_i^u]$, where \mathbf{x}_i^u is the true value of the state of UAV i , then one will have $\mathbb{P}_i((n_r, n_c), \mathbf{x}_i^u) \subset \mathbb{P}_i^{\exists}((n_r, n_c))$.

Therefore, if we manage to characterize $\mathbb{P}_i^{\exists}((n_r, n_c))$, then we can adapt Propositions 9 and 15 by replacing $\mathbb{P}_i((n_r, n_c), \mathbf{x}_i^u)$ with $\mathbb{P}_i^{\exists}((n_r, n_c))$ in order to retain their useful properties even if the UAV state vector is uncertain.

For that purpose, we can characterize an outer-approximation of $\mathbb{P}_i^{\exists}((n_r, n_c))$ using the Thick SIVIA described in Appendix A.4.3. Thus, the set $\mathbb{P}_i^{\exists}((n_r, n_c))$ is the union of the subpavings $\mathcal{B}_{\text{penumbra}}$ and $\mathcal{B}_{\text{border}}$.

In what follows, when a set \mathbb{X} is built using $\mathbb{P}_i((n_r, n_c), \mathbf{x}^u)$, then, the notation $(\mathbb{X})^{\exists}$ means that the set \mathbb{X} is characterized by using an outer-approximation of $\mathbb{P}_i^{\exists}((n_r, n_c))$ instead of $\mathbb{P}_i((n_r, n_c), \mathbf{x}^u)$. The notation $(\mathbb{X})^{\forall}$ would be used to represent that the set \mathbb{X} is characterized using an inner-approximation of $\mathbb{P}_i^{\forall}((n_r, n_c))$ instead of $\mathbb{P}_i((n_r, n_c), \mathbf{x}^u)$.

6.4.2 . Characterization of $\mathbb{P}_i^{\mathcal{G}}((n_r, n_c))$

Assuming the pixel (n_r, n_c) is labeled *Ground*, then $\mathbb{P}_i^{\mathcal{G}}((n_r, n_c), \mathbf{x}^u)$ must be a subset of \mathbb{X}_g that is observed by UAV i for any possible value of its state vector $\mathbf{x}^u \in [\mathbf{x}_i^u]$. According to (6.19), one has $\mathbb{P}_i^{\mathcal{G}, \forall}((n_r, n_c)) \subset \mathbb{P}_i^{\mathcal{G}}((n_r, n_c), \mathbf{x}^u)$ for all $\mathbf{x}^u \in [\mathbf{x}_i^u]$. Consequently, if $\mathbf{x}_i^u \in [\mathbf{x}_i^u]$, where \mathbf{x}_i^u is the true value of the state of UAV i , then one will have $\mathbb{P}_i^{\mathcal{G}, \forall}((n_r, n_c)) \subset \mathbb{P}_i^{\mathcal{G}}((n_r, n_c), \mathbf{x}_i^u)$.

Therefore, if we manage to get an inner-approximation of $\mathbb{P}_i^{\mathcal{G}, \forall}((n_r, n_c))$, then we can adapt Proposition 14 by replacing $\mathbb{P}_i^{\mathcal{G}}((n_r, n_c), \mathbf{x}_i^u)$ with $\mathbb{P}_i^{\mathcal{G}, \forall}((n_r, n_c))$ in order to retain its useful properties even if the UAV state vector is uncertain.

For that purpose, we characterize an inner-approximation of $\mathbb{P}_i^{\mathcal{G}, \forall}((n_r, n_c))$ using the Thick SIVIA described in Appendix A.4.4. Thus, the set $\mathbb{P}_i^{\mathcal{G}, \forall}((n_r, n_c))$ consists only of the subpaving \mathcal{B}_{in} .

As in Section 6.4.1, in what follows, when a set \mathbb{X} is built upon $\mathbb{P}_i^{\mathcal{G}}((n_r, n_c), \mathbf{x}^u)$, then, the notation $(\mathbb{X})^{\forall}$ means that the set \mathbb{X} is characterized by using an inner-approximation of $\mathbb{P}_i^{\mathcal{G}, \forall}((n_r, n_c))$ instead of $\mathbb{P}_i^{\mathcal{G}}((n_r, n_c), \mathbf{x}^u)$. The notation $(\mathbb{X})^{\exists}$ (only useful for $\mathbb{H}_i^{\text{CVS}}$) is used to represent that the set \mathbb{X} is characterized by using an outer-approximation of $\mathbb{P}_i^{\mathcal{G}, \exists}((n_r, n_c))$ instead of $\mathbb{P}_i^{\mathcal{G}}((n_r, n_c), \mathbf{x}^u)$.

Remark 22. One has $\mathbb{P}_i^{\mathcal{G}}(\mathcal{Y}_i^{\mathcal{G}}) = (\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g) \setminus \mathbb{H}_i^{\text{CVS}}$. Therefore, assuming that $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g = \mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$, the characterization of $(\mathbb{P}_i^{\mathcal{G}}(\mathcal{Y}_i^{\mathcal{G}}))^{\forall}$ can be achieved by characterizing $(\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g)^{\forall} = (\mathbb{P}_i^{\mathcal{G}}(\mathcal{N}'))^{\forall}$ and $(\mathbb{H}_i^{\text{CVS}})^{\exists}$. The characterization of $(\mathbb{P}_i^{\mathcal{G}}(\mathcal{N}'))^{\forall}$ is easier to obtain than $\bigcup_{(n_r, n_c) \in \mathcal{N}'} (\mathbb{P}_i^{\mathcal{G}}((n_r, n_c)))^{\forall}$. It is

because $(\mathbb{P}_i^g(\mathcal{N}^l))^\forall$ implicitly involves all the pixels. Thus, its characterization is less likely to be empty than with $(\mathbb{P}_i^g((n_r, n_c)))^\forall$ (the latter being an inner-approximation of the portion of the ground observed by only one pixel instead of the camera). The characterization of $(\mathbb{P}_i^g(\mathcal{N}^l))^\forall$ can be performed with (6.19) but with

$$[\mathbf{y}_{i,r,c}^g] = \begin{pmatrix} [0, d_{\max}] \\ [0, N_c] \\ [0, N_r] \end{pmatrix}. \quad (6.56)$$

The characterization of $(\mathbb{H}_i^{\text{CVS}})^\exists$ can be performed by evaluating the union of the outer-approximations of $\mathbb{P}_i^g((n_r, n_c))$ over all pixels $(n_r, n_c) \in \mathcal{Y}_i^o \cup \mathcal{Y}_i^t \cup \mathcal{Y}_i^n$.

6.5 . Inverse of the measurement function

This section considers the special case where the function \mathbf{h}^{CVS} is bijective, and how the inverse of \mathbf{h}^{CVS} can be used.

6.5.1 . Bijection

Consider a point $\mathbf{x}^{\mathcal{F}_i^c} \in \mathbb{F}(\mathbf{x}_i^u) \setminus \{\mathbf{x}_i^c\}$. According to (3.17), the coordinates (c, r) in the CCD array of the projection of the point $\mathbf{x}^{\mathcal{F}_i^c}$ are

$$\begin{pmatrix} c \\ r \end{pmatrix} = \begin{pmatrix} -f_c & 0 & N_c/2 \\ 0 & -f_r & N_r/2 \end{pmatrix} \begin{pmatrix} x_1^{\mathcal{F}_i^c}/x_3^{\mathcal{F}_i^c} \\ x_2^{\mathcal{F}_i^c}/x_3^{\mathcal{F}_i^c} \\ 1 \end{pmatrix}. \quad (6.57)$$

Therefore, the set of points $\mathbf{x}^{\mathcal{F}_i^c}$ such that $\mathbf{p}_{\mathcal{F}_i^c}(\mathbf{x}^{\mathcal{F}_i^c}) = (c, r)^T$ is

$$\mathbf{p}_{\mathcal{F}_i^c}^{-1}(c, r) = \left\{ \left(x_1^{\mathcal{F}_i^c}, x_2^{\mathcal{F}_i^c}, x_3^{\mathcal{F}_i^c} \right)^T \in \mathbb{R}^2 \times \mathbb{R}^{+*} \mid \begin{pmatrix} (\frac{N_c}{2} - c) \frac{x_3^{\mathcal{F}_i^c}}{f_c} \\ (\frac{N_r}{2} - r) \frac{x_3^{\mathcal{F}_i^c}}{f_r} \\ 1 \end{pmatrix} = \begin{pmatrix} x_1^{\mathcal{F}_i^c} \\ x_2^{\mathcal{F}_i^c} \end{pmatrix} \right\}, \quad (6.58)$$

which is a half-line starting from the origin. Similarly, the set of points $\mathbf{x}^{\mathcal{F}_i^c}$ such that $\|\mathbf{x}^{\mathcal{F}_i^c}\| = \mathbf{D}_i^0(n_r, n_c)$ is the sphere $\mathbb{B}\left((0, 0, 0)^T, \mathbf{D}_i^0(n_r, n_c)\right)$ (not the ball $\mathbb{B}(\cdot, \cdot)$).

Consider a point $\mathbf{x}^* \in \mathbb{F}(\mathbf{x}_i^u) \setminus \{\mathbf{x}_i^c\}$ such that $\mathbf{h}^{\text{CVS}}(\mathbf{x}^*) = (\mathbf{D}_i^0(n_r, n_c), c, r)^T$. This point is the intersection between $\mathbb{B}\left((0, 0, 0)^T, \mathbf{D}_i^0(n_r, n_c)\right)$ and $\mathbf{p}_{\mathcal{F}_i^c}^{-1}(c, r)$. Consequently, the inverse of \mathbf{h}^{CVS} is

$$\mathbf{h}^{\text{CVS}, -1}(\mathbf{y}, \mathbf{x}_i^u) = \frac{\mathbf{D}_i^0(n_r, n_c)}{\nu(c, r)} \mathbf{M}_{\mathcal{F}_i^c}^{\mathcal{F}} \mathbf{M}_{\mathcal{F}_i^c}^{\mathcal{F}^b} \begin{pmatrix} (\frac{N_c}{2} - c) / f_c \\ (\frac{N_r}{2} - r) / f_r \\ 1 \end{pmatrix} + \mathbf{M}_{\mathcal{F}_i^c}^{\mathcal{F}} \mathbf{x}_i^{c, \mathcal{F}_i^b} + \mathbf{x}_i^u, \quad (6.59)$$

with $\mathbf{y} = (\mathbf{D}_i^0(n_r, n_c), c, r)^T$ and

$$\nu(c, r) = \sqrt{\left(\left(\frac{N_c}{2} - c\right) / f_c\right)^2 + \left(\left(\frac{N_r}{2} - r\right) / f_r\right)^2 + 1}. \quad (6.60)$$

6.5.2 . Exploitation of the bijection

According to the Proposition 23, an inclusion function of $(\mathbf{h}^{\text{CVS}})^{-1}$ can be used to characterize an outer-approximation of $\mathbb{P}_i^{\exists}((n_r, n_c))$.

Proposition 23. $\left[(\mathbf{h}^{\text{CVS}})^{-1} \right] \left(\left[\mathbf{y}_{i,r,c}^{\text{t/o}} \right], [\mathbf{x}_i^{\text{u}}] \right) \supset \mathbb{P}_i^{\exists}((n_r, n_c))$

Démonstration. According to (6.23), one has $(\mathbf{h}^{\text{CVS}})^{-1} \left(\left[\mathbf{y}_{i,r,c}^{\text{t/o}} \right], [\mathbf{x}_i^{\text{u}}] \right) \subset \left[(\mathbf{h}^{\text{CVS}})^{-1} \right] \left(\left[\mathbf{y}_{i,r,c}^{\text{t/o}} \right], [\mathbf{x}_i^{\text{u}}] \right)$. Consider $\mathbf{x}^{\text{u}} \in [\mathbf{x}_i^{\text{u}}]$, $(\mathbf{h}^{\text{CVS}})^{-1} \left(\left[\mathbf{y}_{i,r,c}^{\text{t/o}} \right], \mathbf{x}^{\text{u}} \right)$ is the set of the possible points belonging to the shape of the detected entity (target, obstacle,...) while assuming that the state of UAV i is \mathbf{x}^{u} . Thus, one has

$$(\mathbf{h}^{\text{CVS}})^{-1} \left(\left[\mathbf{y}_{i,r,c}^{\text{t/o}} \right], \mathbf{x}^{\text{u}} \right) = \left\{ \mathbf{x} \in \mathbb{X}_0 \mid \exists \mathbf{y} \in \left[\mathbf{y}_{i,r,c}^{\text{t/o}} \right], \mathbf{x} = (\mathbf{h}^{\text{CVS}})^{-1}(\mathbf{y}, \mathbf{x}^{\text{u}}) \right\}. \quad (6.61)$$

Since this is true for all $\mathbf{x}^{\text{u}} \in [\mathbf{x}_i^{\text{u}}]$, one deduces

$$(\mathbf{h}^{\text{CVS}})^{-1} \left(\left[\mathbf{y}_{i,r,c}^{\text{t/o}} \right], [\mathbf{x}_i^{\text{u}}] \right) = \left\{ \mathbf{x} \in \mathbb{X}_0 \mid \exists \mathbf{x}^{\text{u}} \in [\mathbf{x}_i^{\text{u}}], \exists \mathbf{y} \in \left[\mathbf{y}_{i,r,c}^{\text{t/o}} \right], \mathbf{x} = (\mathbf{h}^{\text{CVS}})^{-1}(\mathbf{y}, \mathbf{x}^{\text{u}}) \right\}, \quad (6.62)$$

$$= \mathbb{P}_i^{\exists}((n_r, n_c)). \quad (6.63)$$

Consequently, $\left[(\mathbf{h}^{\text{CVS}})^{-1} \right] \left(\left[\mathbf{y}_{i,r,c}^{\text{t/o}} \right], [\mathbf{x}_i^{\text{u}}] \right) \supset \mathbb{P}_i^{\exists}((n_r, n_c))$. \square

Note that a similar property can be deduced for $\mathbb{P}_i^{\exists}((n_r, n_c))$ but, in that case, we recall that the first component of $\left[\mathbf{y}_{i,r,c}^{\text{g}} \right]$ is $[0, d_{\text{max}}]$ which can be troublesome.

The reason is that $\left[(\mathbf{h}^{\text{CVS}})^{-1} \right] \left(\left[\mathbf{y}_{i,r,c}^{\text{g}} \right], [\mathbf{x}_i^{\text{u}}] \right)$ is a subset of \mathbb{X}_0 (and not \mathbb{X}_{g}) containing all points whose distance from \mathbf{x}_i^{c} is between $[0, d_{\text{max}}]$. This can result in a set that is too large to be useful.

6.6 . Simulation results

This chapter presents an adaptation of the set-membership estimator introduced in Chapter 4 in order to account for the UAV uncertainty. This section presents simulation results related to the characterization of set estimates that are guaranteed to contain or to not contain a target location from CVS measurements obtained at a given time instant. Thus, the time index is omitted in this section.

6.6.1 . Simulation conditions

The simulation conditions are similar to those presented in Chapter 4. Five cars are placed in front of a UAV whose actual state vector (position and attitude only) is $\mathbf{x}_i^{\text{u}} = (0, 0, 50, 0, 0, 0)^{\text{T}}$. The locations of each target are $(60, 0, 0)^{\text{T}}$, $(90, 0, 0)^{\text{T}}$, $(120, 0, 0)^{\text{T}}$, $(150, 0, 0)^{\text{T}}$, and $(180, 0, 0)^{\text{T}}$. Figure 6.2 recalls the considered organization of the targets. The camera produces images that are subsampled to get an image of size $N_r \times N_c = 360 \times 480$ pixels. The bounds of the noise associated to the depth measurement are taken as $[\underline{w}, \bar{w}] = [-1\%, 1\%]$.

To represent the uncertain UAV state, we consider

- (Uncertainty 1) $[\mathbf{w}^{\text{x},1}] = [\pm 0.1 \text{ m}, \pm 0.1 \text{ m}, \pm 0.1 \text{ m}, \pm 0.1 \text{ deg}, \pm 0.1 \text{ deg}]^{\text{T}}$

- (Uncertainty 2) $[\mathbf{w}^{x,2}] = [\pm 5 \text{ m}, \pm 5 \text{ m}, \pm 0.1 \text{ deg}, \pm 0.1 \text{ deg}, \pm 0.1 \text{ deg}]^T$
- (Uncertainty 3) $[\mathbf{w}^{x,3}] = [\pm 0.1 \text{ m}, \pm 0.1 \text{ m}, \pm 0.1 \text{ m}, \pm 0.1 \text{ deg}, \pm 0.1 \text{ deg}, \pm 5 \text{ deg}]^T$
- (Uncertainty 4) $[\mathbf{w}^{x,4}] = [\pm 5 \text{ m}, \pm 5 \text{ m}, \pm 0.1 \text{ m}, \pm 0.1 \text{ deg}, \pm 0.1 \text{ deg}, \pm 5 \text{ deg}]^T$

and $\mathbb{X}_i^{u,\ell} = \mathbf{x}_i^u + [\mathbf{w}^{x,\ell}]$, $\ell = 1, \dots, 4$.

The threshold ε for Thick SIVIA is set to 2 meters. The algorithm stops bisecting a box if all its dimensions are smaller than 2 meters. In addition, we assume that $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g = \mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$, *i.e.*, d_{\max} is not taken into account.

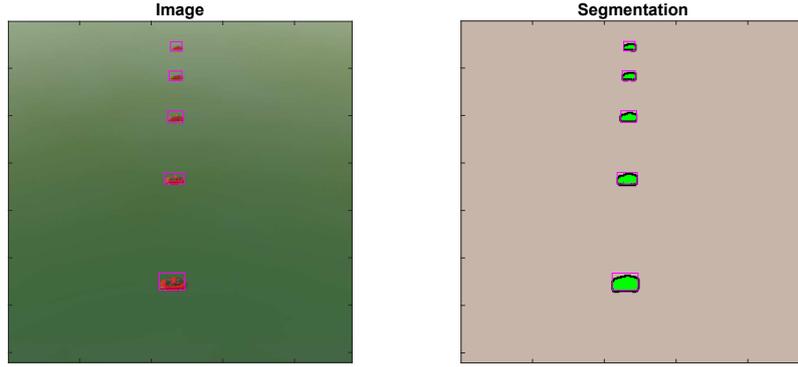


Figure 6.2 – Illustration of some of the CVS measurements that a UAV can acquire in a situation with no obstacles. All targets are identified and the pink boxes represent their location in the image.

6.6.2 . Adaption of the set-membership estimator

This section introduces a change of notation regarding the set estimates $\mathbb{P}_{i,j}^t$, $\mathbb{X}_{i,j}^{t,m}$, $\mathbb{X}_{i,j}^{t,m}$, $\mathbb{X}_{i,j}^t$, \mathbb{X}_i^t , $\mathbb{P}_i^g(\mathcal{Y}_i^g)$, \mathbb{X}_i^o , \mathbb{X}_i^o , and $\mathbb{H}_i^{\text{CVS}}$ to make a clear distinction between the set estimates that are obtained when the UAV state vector is perfectly known, and the set estimates that are obtained from $\mathbb{P}_i^{\exists}((n_r, n_c))$ and $\mathbb{P}_i^{g,\forall}((n_r, n_c))$. The set estimates introduced in Section 4.2 are adapted as follow :

- The union of the $\mathbb{P}_i^{\exists}((n_r, n_c))$ over all the pixels $(n_r, n_c) \in [\mathcal{Y}_{i,j}^t] \cap \mathcal{Y}_i^t$ provides $(\mathbb{P}_{i,j}^t)^{\exists}$
- The characterization from $(\mathbb{P}_{i,j}^t)^{\exists}$ of the set estimate containing the location of the identified target j is denoted $(\mathbb{X}_{i,j}^{t,m})^{\exists}$

The set estimates introduced in Sections 4.3 and 4.4 are adapted as follow :

- The union of the $\mathbb{P}_i^{g,\forall}((n_r, n_c))$ over all the pixels $(n_r, n_c) \in \mathcal{Y}_i^g$ provides $(\mathbb{P}_i^g(\mathcal{Y}_i^g))^{\forall}$
- The union of the $\mathbb{P}_i^{g,\exists}((n_r, n_c))$ over all the pixels $(n_r, n_c) \in \mathcal{Y}_i^o \cup \mathcal{Y}_i^t \cup \mathcal{Y}_i^n$ provides $(\mathbb{H}_i^{\text{CVS}})^{\exists}$
- The characterization from $\mathbb{P}_i^{\exists}((n_r, n_c))$ of the an inner-approximation of the r_s^{to} -ground neighborhood of all the obstacles is denoted $(\mathbb{X}_i^o)^{\exists}$
- The characterization from $(\mathbb{X}_{i,j}^{t,m})^{\exists}$ of an set estimate that cannot contain any target location except $\mathbf{x}_j^{t,g}$ is denoted $(\mathbb{X}_{i,j}^{t,m})^{\exists}$

The set estimates introduced in Sections 4.5 are adapted as follow :

- The set estimate containing all the possible locations of an identified target j is denoted $(\mathbb{X}_{i,j}^t)^{\exists}$

- The set estimate containing all the possible locations of the targets that are not identified yet is denoted $(\overline{\mathbb{X}}_i^t)^\exists$
- The set estimate representing an inner-approximation of the r_s^{to} -ground neighborhood of all the obstacles is denoted $(\mathbb{X}_i^o)^\exists$

6.6.3 . Characterization of a set estimate containing no target location

In this section, we only evaluate the capacity of the proposed approach to characterize the portion of the ground that the UAV has observed while accounting for its uncertain state vector.

First, $\mathbb{X}_i^{u,2}$ has been considered for the uncertain state. Figure 6.3 illustrates $(\mathbb{P}_i^g(\mathcal{N}^l))^\exists$, *i.e.*, the portion of the ground that UAV i may have seen from at least one possible value of its state $\mathbf{x}^u \in \mathbb{X}_i^{u,2}$, and $(\mathbb{P}_i^g(\mathcal{N}^l))^\forall$, *i.e.*, the portion of the ground that UAV i has seen whatever its state $\mathbf{x}^u \in \mathbb{X}_i^{u,2}$. $(\mathbb{P}_i^g(\mathcal{N}^l))^\forall$ is represented by the green set whereas $(\mathbb{P}_i^g(\mathcal{N}^l))^\exists$ is the union of the sets in green, yellow, and blue.

The set $(\mathbb{P}_i^g(\mathcal{N}^l))^\forall$ is contained in the true FoV of the UAV (red), *i.e.*, $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$. This result was to be expected, since $(\mathbb{P}_i^g(\mathcal{N}^l))^\forall$ is an inner-approximation of $\bigcap_{\mathbf{x}^u \in \mathbb{X}_i^{u,2}} (\mathbb{F}(\mathbf{x}^u) \cap \mathbb{X}_g)$. Similarly, it was to be expected that the set $(\mathbb{P}_i^g(\mathcal{N}^l))^\exists$ contains $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$ because $(\mathbb{P}_i^g(\mathcal{N}^l))^\exists$ is an outer-approximation of $\bigcup_{\mathbf{x}^u \in \mathbb{X}_i^{u,2}} (\mathbb{F}(\mathbf{x}^u) \cap \mathbb{X}_g)$. The main difference between $(\mathbb{P}_i^g(\mathcal{N}^l))^\exists$ and $(\mathbb{P}_i^g(\mathcal{N}^l))^\forall$ is that $(\mathbb{P}_i^g(\mathcal{N}^l))^\forall$ is the portion of the ground that UAV i can guarantee to have monitored. The points belonging to $(\mathbb{P}_i^g(\mathcal{N}^l))^\exists \setminus (\mathbb{P}_i^g(\mathcal{N}^l))^\forall$ may have been observed, but no proof is available.

Therefore, when UAV i exploits assumption (3.54) and the pixels labeled *Ground* to characterize an area that cannot contain a target location, it has to characterize two sets. The first one is $(\mathbb{P}_i^g(\mathcal{N}^l))^\forall$ because it represents the portion of the ground that UAV i can guarantee to have monitored. The other set is $(\mathbb{H}_i^{\text{CVS}})^\exists$ because it represents the portion of the ground that may have been masked by a target, an obstacle, or something else. Then, according to Remark 22, $(\mathbb{P}_i^g(\mathcal{Y}_i^g))^\forall = (\mathbb{P}_i^g(\mathcal{N}^l))^\forall \setminus (\mathbb{H}_i^{\text{CVS}})^\exists$.

Figure 6.4 illustrates the set $(\mathbb{P}_i^g(\mathcal{N}^l))^\forall$ that can be obtained for different UAV state uncertainties. The true FoV of UAV i , *i.e.*, $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$, is represented by the set with a black border.

To evaluate the impact of the UAV state uncertainty on the characterization of $(\mathbb{P}_i^g(\mathcal{N}^l))^\forall$, and since $(\mathbb{P}_i^g(\mathcal{N}^l))^\forall \subset (\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g)$, we propose to compute the ratio between the area of $(\mathbb{P}_i^g(\mathcal{N}^l))^\forall$ and the area of $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$, *i.e.*,

$$\phi^{\text{FoV}}(\mathbb{X}_i^{u,\ell}) = \frac{\phi((\mathbb{P}_i^g(\mathcal{N}^l))^\forall)}{\phi(\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g)}. \quad (6.64)$$

With Uncertainty 1, we have $\phi^{\text{FoV}}(\mathbb{X}_i^{u,1}) = 82.2\%$. This means that only 82.2 % of $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$ can be guaranteed to have been monitored. This percentage decreases to 66.2 % with Uncertainty 2, then 63.2 % with Uncertainty 3, and 43, 5 % with Uncertainty 4.

6.6.4 . Fully-visible Target location estimation

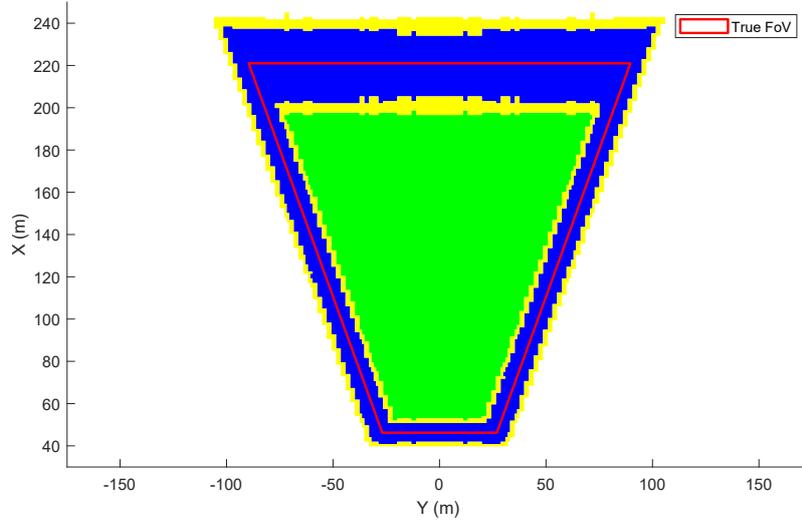


Figure 6.3 – Representation of $(\mathbb{P}_{i,k}^g(\mathcal{N}^l))^{\exists}$ where \mathcal{B}_{in} is in green, $\mathcal{B}_{penumbra}$ is in blue, and \mathcal{B}_{border} is in yellow. The red set is the true intersection of the UAV FoV with the ground when the state is perfectly known.

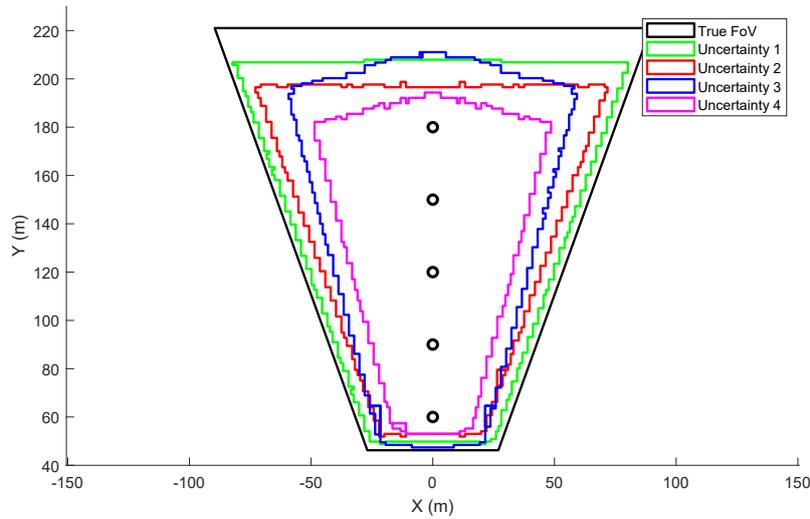


Figure 6.4 – Illustration of $(\mathbb{P}_i^g(\mathcal{N}^l))^{\forall}$ for different values of UAV state uncertainties. The set with a black border is $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$. The black dots represent the 5 target locations.

Figure 6.5 illustrates three different evaluations of the set containing the location of an identified target j . Here, only the target whose location is $(60, 0, 0)^T$ is considered. Uncertainty 4 has been used to characterize the following sets.

The green set is $\mathbb{X}_{i,j}^{t,m}$ obtained without UAV state uncertainty. The set with a red border and the blue set represent $(\mathbb{X}_{i,j}^{t,m})^{\exists}$ where $(\mathbb{P}_{i,j}^t)^{\exists}$ is obtained as the union of the sets $\mathbb{P}_i^{\exists}((n_r, n_c))$ over all pixels $(n_r, n_c) \in [\mathcal{Y}_{i,j}^t] \cap \mathcal{Y}_i^t$. The difference between the set with a red border and the blue set is how

$\mathbb{P}_i^{\exists}((n_r, n_c))$ is characterized. For the set with a red border, $\mathbb{P}_i^{\exists}((n_r, n_c))$ is obtained by only evaluating $\left[(\mathbf{h}^{\text{CVS}})^{-1} \right] \left(\left[\mathbf{y}_{i,r,c}^{\text{t/o}} \right], [\mathbf{x}_i^{\text{u}}] \right)$. Thus, $\mathbb{P}_i^{\exists}((n_r, n_c))$ is only one box. For the blue set, $\left[(\mathbf{h}^{\text{CVS}})^{-1} \right] \left(\left[\mathbf{y}_{i,r,c}^{\text{t/o}} \right], [\mathbf{x}_i^{\text{u}}] \right)$ is used as the starting box for Thick SIVIA in order to get a smaller outer-approximation of $\mathbb{P}_i^{\exists}((n_r, n_c))$. The green set is only here for the sake of size comparison.

The blue set is (a little) smaller than the set with a red border. This result was to be expected since we use Thick SIVIA to remove a portion of $\left[(\mathbf{h}^{\text{CVS}})^{-1} \right] \left(\left[\mathbf{y}_{i,r,c}^{\text{t/o}} \right], [\mathbf{x}_i^{\text{u}}] \right)$ that are not in $\mathbb{P}_i^{\exists}((n_r, n_c))$. Nevertheless, the set with a red border is obtained in 0.08 second, whereas the blue set is obtained in 119.03 ± 0.36 seconds¹. This drastic increase in computation time is because UAV i has to run a 3D Thick SIVIA to get an outer approximation of $\mathbb{P}_i^{\exists}((n_r, n_c))$ for every pixel $(n_r, n_c) \in \left[\mathcal{Y}_{i,j}^{\text{t}} \right] \cap \mathcal{Y}_i^{\text{t}}$, which is around 400 pixels in this case.

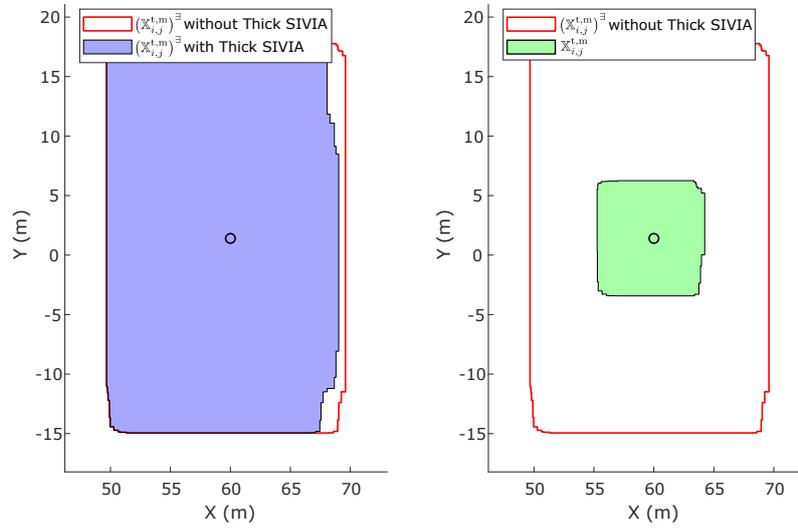


Figure 6.5 – Comparison between two different approaches to get an outer-approximation of $\left(\mathbb{X}_{i,j}^{\text{t,m}} \right)^{\exists}$. The set with a red border is obtained by an approach that does not use Thick SIVIA whereas the blue set is obtained with an approach that uses Thick SIVIA. The green set represents the set $\mathbb{X}_{i,j}^{\text{t,m}}$ that would be obtained if there are no UAV state uncertainties. The black dot represents the target location.

Figure 6.6 compares the set estimates that are would be obtained without UAV state uncertainties, with the set estimates that are obtained when Uncertainty 1 is considered. The set estimates $\mathbb{P}_i^{\mathcal{G}}(\mathcal{Y}_i^{\mathcal{G}})$ and $\left(\mathbb{P}_i^{\mathcal{G}}(\mathcal{Y}_i^{\mathcal{G}}) \right)^{\forall}$ are represented by the blue sets, $\mathbf{p}_g \left(\mathbb{P}_{i,j}^{\text{t}} \right)$ and $\left(\mathbf{p}_g \left(\mathbb{P}_{i,j}^{\text{t}} \right) \right)^{\exists}$ are represented by the sets with a red border, and $\mathbb{X}_{i,j}^{\text{t,m}}$ and $\left(\mathbb{X}_{i,j}^{\text{t,m}} \right)^{\exists}$ are represented with the sets with a green border. The first line of Figure 6.6 is dedicated to the set estimates obtained without UAV state uncertainty, and the second line is dedicated to the set estimates obtained while taking into account the UAV state uncertainty.

One can remark that, even though $\mathbb{X}_{i,j}^{\text{t,m}}$ and $\left(\mathbb{X}_{i,j}^{\text{t,m}} \right)^{\exists}$ look similar, $\mathbb{P}_i^{\mathcal{G}}(\mathcal{Y}_i^{\mathcal{G}})$ and $\left(\mathbb{P}_i^{\mathcal{G}}(\mathcal{Y}_i^{\mathcal{G}}) \right)^{\forall}$ are not

1. Those results have been averaged over 10 independent simulations run on a computer equipped with a Intel(R) Core(TM) i7-10875H 2.30GHz

the same. The reason lies in the characterization of $(\mathbb{H}_i^{CVS})^\exists$. Due to the UAV state uncertainties, UAV i must evaluate an outer-approximation of the union of the portions of the ground that may have been masked by a target for each possible value of its state vector. Thus, $(\mathbb{H}_i^{CVS})^\exists$ is larger. Consequently, even though $\mathbb{X}_{i,j}^{t,m}$ and $(\mathbb{X}_{i,j}^{t,m})^\exists$ are almost the same, a smaller portion of $(\mathbb{X}_{i,j}^{t,m})^\exists$ can be removed by using $(\mathbb{P}_i^g(\mathcal{Y}_i^g))^\forall$ to reduce the localization uncertainty.

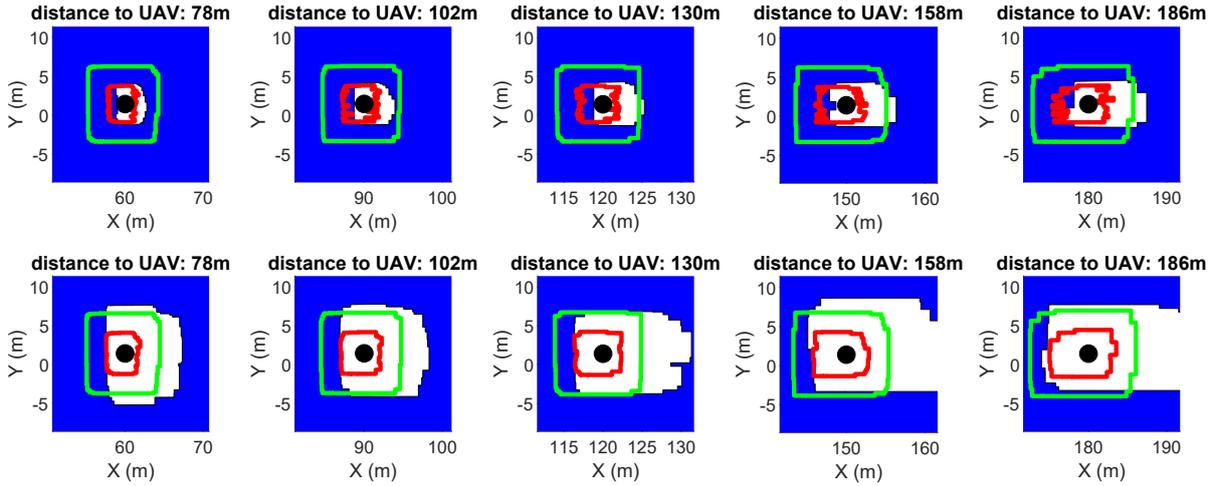


Figure 6.6 – Comparison of the set estimates that would be obtained without UAV state uncertainties (first line), with the set estimates that are obtained while considering an uncertain UAV state vector (second line). For both lines, $\mathbb{P}_i^g(\mathcal{Y}_i^g)$ and $(\mathbb{P}_i^g(\mathcal{Y}_i^g))^\forall$ are represented by the blue sets, $\mathbf{p}_g(\mathbb{P}_{i,j}^t)$ and $(\mathbf{p}_g(\mathbb{P}_{i,j}^t))^\exists$ are represented with by the sets with a red border, and $\mathbb{X}_{i,j}^{t,m}$ and $(\mathbb{X}_{i,j}^{t,m})^\exists$ are represented with the sets with a green border. The black dots represent the target locations.

Figure 6.7 illustrates the performance of the set-membership estimator to provide an accurate estimation of the target location while accounting for UAV state uncertainties (Uncertainty 1) (right part of the figure). This performance is compared with the performance of the same estimator, but when the UAV state vector is precisely known (left part of the figure).

As we expected with Figure 6.6, $\phi^{eR}(\mathbb{X}_{i,j}^{t,m})$ and $\phi^{eR}((\mathbb{X}_{i,j}^{t,m})^\exists)$ are almost the same. Nonetheless, when UAV i characterizes the set estimate $(\mathbb{X}_{i,j}^{t,m})^\exists$ during the correction step, *c.f.*, (4.32) (or (4.46) for a simplified version of the correction), since $(\mathbb{P}_i^g(\mathcal{Y}_i^g))^\forall$ is smaller than $\mathbb{P}_i^g(\mathcal{Y}_i^g)$, a smaller portion of $(\mathbb{X}_{i,j}^{t,m})^\exists$ is removed. Thus, as indicated by the metric $\phi^{eR}((\mathbb{X}_{i,j}^{t,m})^\exists)$ (the black curve in the right subfigure), the localization uncertainty is not reduced as much as with $\phi^{eR}(\mathbb{X}_{i,j}^{t,m})$ (the black curve in the left subfigure). For instance, if we only consider the target whose location is $(60, 0, 0)^\top$, *i.e.*, its distance to the UAV is around 78 m, the localization uncertainty without UAV state uncertainty is around 2.39 m whereas the localization uncertainty when an uncertain UAV state vector is considered is 4.66 ± 0.49 m

Strangely, the distance between the actual target location and the barycenter of $\left(\mathbb{X}_{i,j}^t\right)^\exists$, *i.e.*, $\phi^c\left(\left(\mathbb{X}_{i,j}^t\right)^\exists\right)$ (in red), is smaller when we consider an uncertain state vector. The reason behind this phenomenon is that $\left(\mathbf{p}_g\left(\mathbb{P}_{i,j}^t\right)\right)^\exists$ contains $\mathbf{x}_j^{t,g}$ because the UAV is able to get some measurements from the sunroof of the cars. Consequently, $\left(\mathbf{p}_g\left(\mathbb{P}_{i,j}^t\right)\right)^\exists$ is likely to be centered around $\mathbf{x}_j^{t,g}$, and since a small portion of $\left(\mathbb{X}_{i,j}^{t,m}\right)^\exists$ is removed by $\left(\mathbb{P}_i^g\left(\mathcal{Y}_i^g\right)\right)^\forall$, $\left(\mathbb{X}_{i,j}^t\right)^\exists$ is also likely to be centered around $\mathbf{x}_j^{t,g}$.

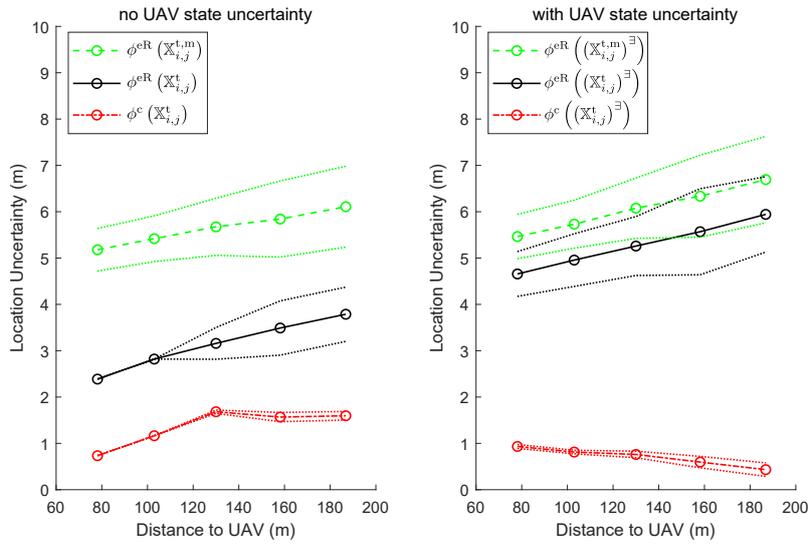


Figure 6.7 - On the left : Mean value and standard deviation over 100 simulations of $\phi^{eR}\left(\mathbb{X}_{i,j}^{t,m}\right)$ (in green), $\phi^{eR}\left(\mathbb{X}_{i,j}^t\right)$ (in black), and $\phi^c\left(\mathbb{X}_{i,j}^t\right)$ (in red). The UAV state vector is assumed perfectly known. On the right : Mean value with standard deviation over 10 simulations of $\phi^{eR}\left(\left(\mathbb{X}_{i,j}^{t,m}\right)^\exists\right)$ (in green), $\phi^{eR}\left(\left(\mathbb{X}_{i,j}^t\right)^\exists\right)$ (in black), and $\phi^c\left(\left(\mathbb{X}_{i,j}^t\right)^\exists\right)$ (in red). Uncertainty 1 is considered to model the UAV state uncertainties.

Figure 6.8 illustrates the impact that an uncertain UAV state vector has regarding the usefulness of assumption 3.11.

With Proposition 18, UAV i can exploits $\mathbb{X}_{i,j}^{t,m}$ to characterize a set estimate $\underline{\mathbb{X}}_{i,j}^{t,m}$ that cannot contain the location of any target except $\mathbf{x}_j^{t,g}$. In Chapter 4, we have shown with Figure 4.19 that this proposition can be used to prove that almost all the ground that is hidden behind a target cannot contain the location of another target. This result is useful to further reduce $\overline{\mathbb{X}}_{i,k}^t$.

Nonetheless, as illustrated in Figure 6.8, the characterization of $\left(\underline{\mathbb{X}}_{i,j}^{t,m}\right)^\exists$ from $\left(\mathbb{X}_{i,j}^{t,m}\right)^\exists$, see Appendix A.2, is almost useless, even with $r_s^{tt} = 8$ m. There are two reasons for this. The first one is that the characterization of $\left(\underline{\mathbb{X}}_{i,j}^{t,m}\right)^\exists$ relies on an inner-approximation of the intersection of all the discs of ra-

dus r_s^{tt} which have their center in $(\underline{\mathbb{X}}_{i,j}^{t,m})^\exists$. Consequently, since $(\underline{\mathbb{X}}_{i,j}^{t,m})^\exists$ is a little bigger than $\underline{\mathbb{X}}_{i,j}^{t,m}$, then this intersection is either smaller or empty. The second reason is that $(\mathbb{H}_i^{CVS})^\exists$ is larger. Therefore, if we have $(\mathbb{P}_i^g(\mathcal{Y}_{i,j}^t))^\exists \subset (\underline{\mathbb{X}}_i^t)^\exists$, i.e., the portion of the ground masked by target j may contain an unidentified target, then $(\underline{\mathbb{X}}_{i,j}^{t,m})^\exists$ must be even larger than $\underline{\mathbb{X}}_{i,j}^{t,m}$ to safely remove $(\mathbb{P}_i^g(\mathcal{Y}_{i,j}^t))^\exists \cap (\underline{\mathbb{X}}_i^t)^\exists$. That will not be the case. In fact $\Phi^\Delta \left((\underline{\mathbb{X}}_{i,j}^{t,m})^\exists, (\underline{\mathbb{X}}_i^t)^\exists \cap (\mathbb{P}_i^g(\mathcal{Y}_{i,j}^t))^\exists \right)$ would be even smaller than $\Phi^\Delta \left(\underline{\mathbb{X}}_{i,j}^{t,m}, \underline{\mathbb{X}}_{i,k}^t \cap \mathbb{P}_i^g(\mathcal{Y}_{i,j}^t) \right)$, where

$$\Phi^\Delta(\mathbb{X}_1, \mathbb{X}_2) = 100\phi(\mathbb{X}_1 \cap \mathbb{X}_2) / \phi(\mathbb{X}_1) \quad (6.65)$$

evaluates how much \mathbb{X}_2 overlaps \mathbb{X}_1 .

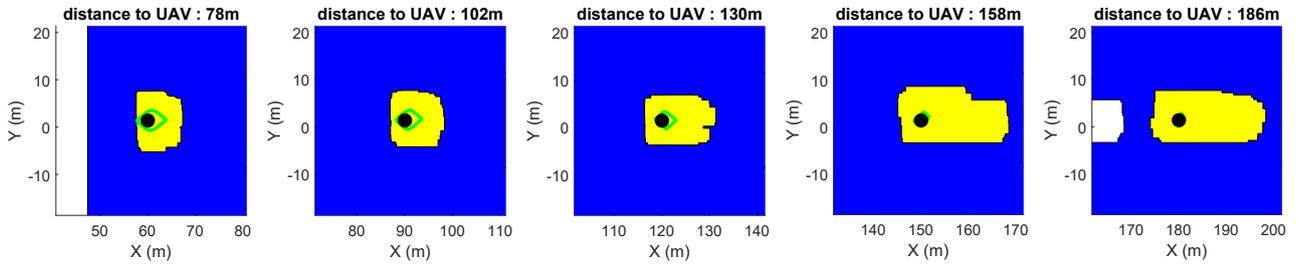


Figure 6.8 – Representation of the set estimates $(\mathbb{P}_i^g(\mathcal{Y}_{i,j}^t))^\exists$ (in blue), $(\underline{\mathbb{X}}_i^t)^\exists \cap (\mathbb{P}_i^g(\mathcal{Y}_{i,j}^t))^\exists$ (in yellow), and $(\underline{\mathbb{X}}_{i,j}^{t,m})^\exists$ (in green). The black dots are the different target locations. Here, $r_s^{tt} = 8$ m

6.6.5 . Inner approximation of the r_s^{to} -ground neighborhood of an obstacle

Figure 6.9 provides some results regarding the capacity of a UAV with an uncertain state vector to obtain an inner-approximation of the r_s^{to} -ground neighborhood of an obstacle. Different values of r_s^{to} are considered. The UAV state uncertainties that are considered for that figure are :

- $[\mathbf{w}^x] = [\pm 0.1 \text{ m}, \pm 0.1 \text{ m}, \pm 0.1 \text{ m}, \pm 0.1 \text{ deg}, \pm 0.1 \text{ deg}, \pm 0.1 \text{ deg}]^T$ (Fig A)
- $[\mathbf{w}^x] = [\pm 1 \text{ m}, \pm 1 \text{ m}, \pm 0.1 \text{ m}, \pm 0.1 \text{ deg}, \pm 0.1 \text{ deg}, \pm 0.1 \text{ deg}]^T$ (Fig B)
- $[\mathbf{w}^x] = [\pm 2 \text{ m}, \pm 2 \text{ m}, \pm 0.1 \text{ m}, \pm 0.1 \text{ deg}, \pm 0.1 \text{ deg}, \pm 0.1 \text{ deg}]^T$ (Fig C)

In Figure 6.9-A, the UAV is able to characterize $(\underline{\mathbb{X}}_i^o)^\exists$ for each r_s^{to} . Nonetheless, as the UAV state uncertainties increase, the UAV has more difficulties in characterizing the r_s^{to} -ground neighborhood of close obstacles. This is even more the case when we look at Figure 6.9-C. The UAV state uncertainty is mainly about the UAV location, and we only considered an uncertain location of ± 2 m. Nonetheless, even with this relatively small UAV location uncertainty, the UAV is unable to characterize $(\underline{\mathbb{X}}_i^o)^\exists$ for small values of r_s^{to} . This is a problem because, since we cannot assume a too large safety distance between the targets and the obstacles, if the UAV is unable to characterize $(\underline{\mathbb{X}}_i^o)^\exists$ even for close obstacles, then it cannot gain environmental knowledge during its exploration of the RoI. Consequently, the performance of the implemented strategy to solve the CSAT problem is reduced, c.f., Chapter 5.

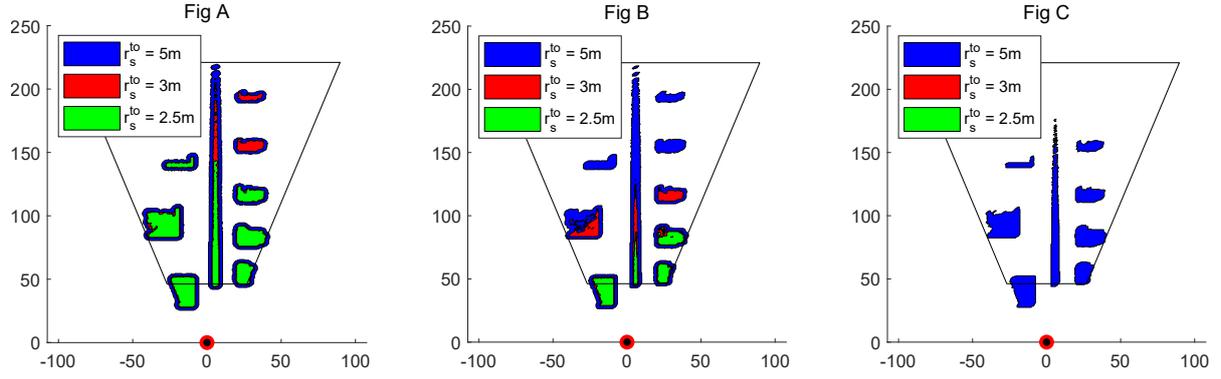


Figure 6.9 – Comparison of $(\mathbb{X}_i^0)^\exists$ for different values of r_s^{to} and UAV state uncertainties. The red dot represents the actual location of the UAV and the set with a black border represents $\mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_g$.

6.7 . Conclusion

This last chapter considers that the UAVs tasked to search for an unknown number of targets within an unknown environment no longer have access to an accurate estimation of their state vector. Thus, when a UAV processes its available CVS measurements to update the set estimates that are guaranteed to contain the location of a target, it has to account for its uncertain state vector. This is because the CVS measurements are initially expressed in a frame attached to that UAV whereas the resulting set estimates are required to be expressed in the reference frame \mathcal{F} . Consequently, the target location estimator introduced in Chapter 4 must be adapted to be robust against the UAV state uncertainties.

Nonetheless, the set estimates characterized and used by the target location estimator are all built on the set estimates $\mathbb{P}_{i,k}((n_r, n_c))$ and $\mathbb{P}_{i,k}^g((n_r, n_c))$. Consequently, we do not need to adapt the entire algorithm but only the characterization of $\mathbb{P}_{i,k}((n_r, n_c))$ and $\mathbb{P}_{i,k}^g((n_r, n_c))$. This chapter presents an approach to characterize a new version of these set estimates, denoted $\mathbb{P}_{i,k}^\exists((n_r, n_c))$ and $\mathbb{P}_{i,k}^{g,\forall}((n_r, n_c))$, that carry the same useful properties of their predecessors while accounting for an uncertain UAV state vector. The method used to characterize $\mathbb{P}_{i,k}^\exists((n_r, n_c))$ and $\mathbb{P}_{i,k}^{g,\forall}((n_r, n_c))$ relies on interval analysis. The characterization of $\mathbb{P}_{i,k}^\exists((n_r, n_c))$ and $\mathbb{P}_{i,k}^{g,\forall}((n_r, n_c))$ can be determined by solving a set inversion problem. Thus, SIVIA may be used to approximate these set estimates with a paving of non-overlapping boxes. Nonetheless, since this set inversion problem involves an uncertain vector of parameters (here : the UAV state vector), SIVIA is too computationally expensive. Instead, we use Thick SIVIA to solve this problem efficiently. To use Thick SIVIA, we introduce a function that relates a point of the RoI to a depth measurement and the coordinates of a pixel containing the projection of that point in the CCD array while making the dependency explicit in the UAV state vector. A method to obtain the thick inclusion function of the measurement function is presented. Then, Thick SIVIA uses this thick inclusion function to provide either an inner-approximation of $\mathbb{P}_{i,k}^{g,\forall}((n_r, n_c))$ or an outer-approximation of $\mathbb{P}_{i,k}^\exists((n_r, n_c))$ more efficiently than with SIVIA.

Several simulations have been performed to evaluate the performance of the target location estimator when it has to be robust to UAV state uncertainties. The approach is still able to characterize

from CVS measurements a set estimate that is guaranteed to contain the location of an identified target. Nevertheless, the accuracy of the estimated target location is clearly reduced. There are two possible reasons for this loss of performance. The first one is, since the UAV state vector is uncertain, $\left(\mathbb{X}_{i,j,k}^{t,m}\right)^\exists$ is larger than $\mathbb{X}_{i,j,k}^{t,m}$. Then, the UAV has some difficulties in exploiting negative information, *i.e.*, the presence of an obstacle, or the detection of the ground, to characterize set estimates that are guaranteed never to contain a target location. For instance, in Figure 6.9, the UAV may be unable to exploit pixels labeled *Obstacle* to characterize a non-empty set around the detected obstacles such that it never contains a target location. Additionally, the use of Thick SIVIA to process CVS measurements might be a bad approach. According to Figure 6.5, the characterization of $\left(\mathbb{X}_{i,j,k}^{t,m}\right)^\exists$ involves a sequential execution of Thick SIVIA in a 3D space to get $\mathbb{P}_{i,k}^\exists((n_r, n_c))$ for each *Target*-labeled pixels $(n_r, n_c) \in \mathcal{Y}_{i,k}^t \cap \left[\mathcal{Y}_{i,j,k}^t\right]$. Since $\mathcal{Y}_{i,k}^t \cap \left[\mathcal{Y}_{i,j,k}^t\right]$ may contain hundreds or thousands of pixels, the characterization of $\left(\mathbb{X}_{i,j,k}^{t,m}\right)^\exists$ would be too demanding for real-time applications.

Chapter 7 - Conclusion

This thesis addresses the problem of searching, identifying, localizing, and tracking an unknown number of mobile ground targets in an unknown and cluttered Region of Interest (RoI) using a fleet of cooperative UAVs. Each UAV embeds a Computer Vision System (CVS) consisting of a camera and image processing algorithms. The CVS processes the information collected from the perceived environment in order to provide some measurements such as a labeled image, a depth map, or a bounding box when a target is identified. These CVS measurements are related to the presence or the absence of a target or an obstacle and, thus, can be exploited to solve a localization problem in order to estimate the possible locations of the targets. Once the CVS measurements available at a given time have been processed, a UAV exploits the updated knowledge about the possible locations of the identified and unidentified targets as well as its knowledge about the environment, *i.e.*, the location and shape of the obstacles, to update its trajectory. Moreover, if the UAV can communicate with its neighbors, then it can update its own knowledge and its trajectory in a cooperative fashion.

To address this problem, formalized as a Cooperative Search, Acquisition, and Tracking (CSAT) problem, we consider a set-membership approach. Hence, we model the measurement noise and other uncertainties as bounded with known bounds. This allows to characterize the set estimates that are guaranteed to contain the location of the targets, provided that the hypotheses on the measurement models and noise bounds are satisfied. An alternative approach would be to consider stochastic assumptions on the measurement noises. Nonetheless, these assumptions, *i.e.*, the known bounds for the set-membership approach or the knowledge of the probability density functions for the stochastic approach, are difficult to justify. Moreover, the CVS embedded in each UAV often involves deep learning algorithms. Consequently, it is difficult to get measurement models and to characterize the measurement noise of these algorithms since their performance highly depends of the training dataset on which they were trained.

To address these issues, we proposed in Chapter 3 a formalization of the CSAT problem. This formalization includes assumptions about the structured unknown RoI, the targets, the UAVs, and the embedded CVS. For instance, the embedded camera is modeled by a simple and common geometric model, *i.e.*, the pinhole model without distortion. Regarding the CVS, several assumptions and models have been introduced to describe how the measurements provided by the CVS can be related to the environment, the targets, and the obstacles. The validity of these assumptions and models is arguable, depending on specific scenario at hand, but they are mild enough to model the information contained in the CVS measurements that can be used to localize the targets. Among the proposed assumptions, the one that has been favored in this thesis is that if a pixel is labeled either *Target*, *Obstacle*, or *Ground*, then that pixel is assumed to be illuminated by light rays stemming only from a target, an obstacle, or the ground respectively. Besides, *Ground*-labeled pixels are used along a reasonable observation condition between a UAV and a target in order to get a negative measurement, *i.e.*, a measurement that can be used to prove the absence of a target. Additionally, even though this thesis considers only an RGB camera to collect information from the environment, the proposed framework can be extended to include other sensors and algorithms that have not been covered in the thesis, such as thermal cameras.

These assumptions and models are then used to solve a localization problem. Chapter 4 presents the first (to the best of our knowledge) set-membership target location estimator exploiting directly multiple measurements provided by a CVS. The proposed estimator exploits the CVS measurements related to the presence of an identified target to characterize a set estimate that is guaranteed to contain the location of that target. The estimator also exploits CVS measurements related to the obstacles and the ground to characterize two set estimates that are guaranteed not to contain the location of a target. Once a UAV has characterized these sets, it updates its previous knowledge regarding the potential locations of identified and unidentified targets through a prediction-correction process. Additionally, if the UAVs have enough computational power, then the CVS measurements can be used to map the unknown RoI using a 2.5D mapping approach, *i.e.*, with an Occupancy-Elevation Map. The purpose of this map is mainly to get a rough approximation of the location and shapes of the *a priori* unknown obstacles. Several simulations have been performed to evaluate the performance of the target location estimator by only considering measurements obtained at a given time instant, and without *a priori*. These simulations show that the estimated target location remains accurate (under 2 m), even when the depth measurement noise is large, or when the *a priori* known target dimensions are inaccurate.

Once a UAV has updated its knowledge by exploiting the CVS measurements and the information broadcast by its neighbors, it updates its trajectory to keep searching for new targets while maintaining an accurate estimate of the location of the identified targets. Nonetheless, since the targets are assumed to outnumber the UAVs, a trade-off between searching for new targets and pursuing already identified targets to maintain an accurate estimation of their location must be found. To address this problem, we consider a control design based on a distributed MPC. Chapter 5 presents this control strategy. Our aim is to update the trajectory of a UAV such that it minimizes the uncertainty of the estimated location of both identified and unidentified targets. Thus, a criterion that translates this objective is introduced, and the MPC searches for the sequence of control inputs that minimizes this criterion. In this chapter, the proposed control strategy is built upon the one presented in (Ibenthal, 2022). In fact, the proposed simulations evaluate the performance of the criterion proposed in (Ibenthal, 2022) but with 3 different levels of knowledge regarding the environment. The first level, *i.e.*, the one considered in (Ibenthal, 2022), assumes that no environmental knowledge is available to evaluate the impact of a sequence of control inputs on the reduction of uncertainty. This first level leads to trajectories that do not account for the presence of obstacles and, thus, leads the UAVs towards locations where obstacles may occupy a huge part of their field of view. The second level considers that the knowledge acquired from the obstacles that are currently under observation can be used to better evaluate the impact of a sequence of control inputs on the reduction of uncertainty. This level enables the UAVs to take the presence of some obstacles into account, while evaluating how they may reduce their field of view in the future. Unfortunately, since there is no mapping, this approach cannot predict the impact of the obstacles that are not currently under observation. To address this problem, a third level is considered. This level takes advantage of a 2.5D map to predict the portion of the RoI that a UAV would not observe due to the presence of a previously detected obstacle. This prediction of the hidden areas is used to search for the sequence of control inputs that lead to interesting points of view to observe an area that may contain a target while avoiding obstacle occlusions. Simulations in an urban environment have shown that this third level always provides the best target search and

tracking performance. In addition, a new criterion is proposed to enhance the target search within a cluttered environment. Nonetheless, the simulations show that this improvement in target search comes at the cost of a slight decrease in the capacity of the UAVs to maintain an accurate estimation of the location of identified targets.

Chapter 6 presents the final work of the thesis. It improves the target location estimator presented in Chapter 4 in order to make it robust to the state uncertainty of the UAVs. The formalism introduced in Chapter 3 is unchanged. Only the method to characterize the set estimates that either are guaranteed to contain the location of a target or are guaranteed to not contain a target location is adapted. This adaptation is done by using interval analysis and thick intervals in order to solve this problem of robustness as a set inversion problem. Nevertheless, even though the simulations show that the proposed adaptation contributes to robust target location estimates, the proposed method requires too much time to be used in an on-board system.

Chapter 8 - Perspectives

This chapter presents some research directions that may extend the work presented in this thesis. Each of the following sections highlights a limit of the assumptions, models, and algorithms that have been considered. Then, different ideas are proposed to address these problems while providing some insights on the difficulties that may be encountered.

8.1 . Imperfect target detection and identification

In Section 3.5, we introduced some assumptions and models to describe the way measurements provided by the CVS can be related to the environment. Those CVS measurements consist of a depth map, an RGB image, a classified image, and a set of bounding boxes locating the identified target in the image. We assumed that the depth map has the same resolution as the image. This can be the case depending on the type of CVS used, see Section 2.2.

Regarding the image segmentation, we considered, according to (Jiang et al., 2023), that the pixel classifier is able to evaluate the confidence it has on attributing one of the three considered labels, *i.e.*, Ground, Target, or Obstacle, to a pixel. If the confidence is too low, then the pixel is considered Not-labeled and, thus, not used in most of the presented algorithms.

We also consider that a target detector is used. A part of this information is a bounding box that correctly localizes the target in the image. Using (3.53), the imperfection of target detectors, *i.e.*, with a bad IoU (Intersection over Union) score, can be handled. Those imperfections may come from a poor training dataset regarding the actual scenario, or a target located in a crowded place. Then, based on the algorithm presented in (Oron et al., 2017), we assumed that as soon as a bounding box is available the target is identified. Nevertheless, we do not account for target misidentification.

8.1.1 . Proposed approach

Trees have not been considered in this thesis because (i) they are not Z -convex and (ii) Webots have some issues in generating correct CVS measurements when a tree is within the FoV of a UAV. Regardless of this problem, one would be interested in using the *Tree*-labeled pixels $\mathcal{Y}^{\text{tree}}$ to characterize and predict areas where the UAV would have more difficulties to detect/identify a target. This additional environmental knowledge may be represented by a set like $\underline{\mathbb{X}}_{i,k}^{\text{o}}$, or another OEM to predict (and avoid) these areas occluded by a tree during the trajectory update.

Regarding the target identification, two problems can be considered :

- how to model the capacity of a UAV to identify a target, and with which confidence
- how to manage non-identification and misidentification

Regarding the first problem, one may assume to have access to $\mathcal{Y}_{i,j,k}^{\text{t}}$ and compute the ratio

$$\text{card} (\mathcal{Y}_{i,j,k}^{\text{t}} \cap [\mathcal{Y}_{i,j,k}^{\text{t}}]) / \text{card} (\mathcal{Y}_{i,k}^{\text{t}} \cap [\mathcal{Y}_{i,j,k}^{\text{t}}]) \quad (8.1)$$

to estimate, with a given threshold, if the target has been identified or not. Otherwise, one may use the geometric approach developed in (Ibenthal et al., 2023), or the stochastic approach presented

in (Meera et al., 2019). Another possible approach is the following : if the UAV has detected but not identified target j , with $j \in \mathcal{L}_{i,k-1}^t$, then if $\mathbb{F}(\mathbf{x}_{i,k}^u) \cap \mathbb{X}_g$ has a non-empty intersection with $\mathbb{X}_{i,j,k|k-1}^t$ and an empty intersection with $\overline{\mathbb{X}}_{i,k|k-1}^t$ and $\mathbb{X}_{i,\ell,k|k-1}^t$ for $\ell \in \mathcal{L}_{i,k-1}^t \setminus \{j\}$, that one can conclude that the only possible target in the sensing range is target j

However, non-identification is not a problem with our approach because, even if target j is not identified, the projection of its location on the image, *i.e.*, $\mathbf{p}_c(\mathbf{x}_{i,k}^u, \mathbf{x}_{j,k}^{t,g})$, cannot belong to a Ground-labeled pixel, see Section 3.6 of Chapter 3. Therefore, the location of a non-identified target j cannot be removed from $\mathbb{X}_{i,j,k|k-1}^t$ or $\overline{\mathbb{X}}_{i,k|k-1}^t$ by $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$.

For misidentification, one can take inspiration from (Ibenthal et al., 2020) to improve the target location estimator. Additionally, in (Zagar et al., 2022) a robust approach has been developed (not presented in this thesis) to estimate the location of a target from a list of set estimates where some of them are outliers, *i.e.*, there is a target misidentification or the measurement noise bounds have been violated. This list is the collection of set estimates provided by neighboring UAVs, *e.g.*, $\mathbb{X}_{i,j,k|k-1}^t$, $\mathbb{X}_{i,j,k}^{t,m}$ and $\mathbb{X}_{\ell,j,k}^{t,m}$ with $\ell \in \mathcal{N}_{i,k} \setminus \{i\}$. Then, instead of using the basic intersection, as in Section 4.5.3 of Chapter 4, one can use the q -relaxed intersection to obtain a set that has a high chance of containing the target location while identifying which UAV has generated the outliers. Figure 8.1 illustrates different set estimate intersections while considering an outlier.

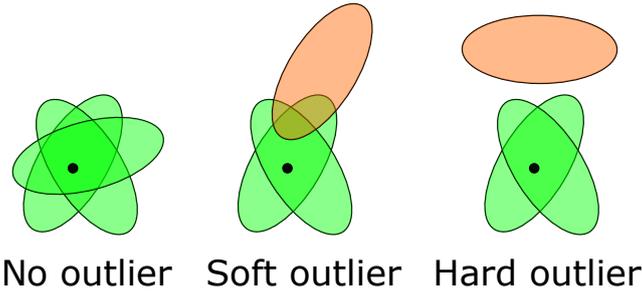


Figure 8.1 – Estimation of the location of a target (black dot) from several set estimates. The green sets are the set estimates that are guaranteed to contain the target location. The orange set estimate represents an outlier. The left part shows a situation where no outlier is present. The middle part shows a situation with one outlier. However, since the intersection of the 3 sets is not empty, the error cannot be detected. The right part shows a similar situation but where the intersection of the 3 sets is empty. The presence of the outlier can be detected and rejected using q -relaxed intersection.

8.2 . Uneven ground

One of our assumptions regarding the environment was to assume that the ground \mathbb{X}_g is a flat horizontal plane of known altitude. This assumption has been used to characterize $\mathbb{P}_{i,k}^g((n_r, n_c))$. This set is used to either characterize portions of the ground that are free of obstacles, *i.e.*, $\mathbb{P}_{i,k}^g(\mathcal{Y}_{i,k}^g)$ without accounting for the depth map, or approximate portions of the ground that are hidden behind an obstacle $\mathbb{H}_{i,k}^{CVS}$. Additionally, \mathbb{X}_g allowed to evaluate $\mathbb{H}_{i,k}^{OEM}(\mathbf{x}_{i,k+\tau|k}^{u,P})$, with $\tau = 1, \dots, h$. With an

unknown uneven ground \mathbb{X}_g , the UAV is no longer able to get those sets.

8.2.1 . Proposed approach

As illustrated in Figure 8.2, Webots is able to create an uneven ground. Thus, a new challenge is how the UAVs can gain knowledge regarding the ground from CVS measurements.

One can adapt the proposed approach to approximate the location and shape of the obstacles using the OEM to approximate the shape of the ground. With this approach, *empty* cells of the OEM will no longer have an elevation of 0. Instead, $\mathbb{P}_{i,k}((n_r, n_c))$ can be used for each Ground-labeled pixel to estimate locally the height of the ground, and update the map accordingly. The OEM is then used to approximate the uneven ground in order to characterize $\mathbb{P}_{i,k}^g((n_r, n_c))$ or $\mathbb{H}_{i,k}^{\text{OEM}}$.

Nonetheless, assuming an unstructured uneven ground may lead to situations where Assumption 3.54 is no longer verified. This assumption is mandatory for the proposed target location estimator since it allows to characterize a set that is guaranteed to contain no target location. Consequently, to apply the target location estimator detailed in this thesis, one may have to assume a structured uneven ground.

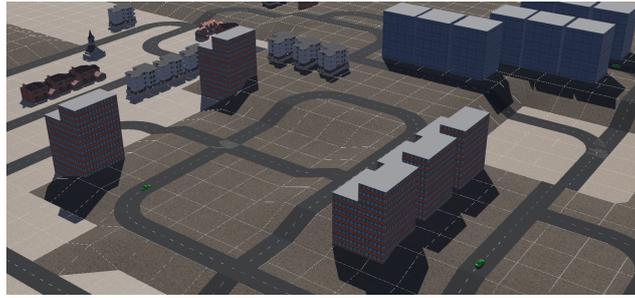


Figure 8.2 – Illustration of the city setup with uneven ground created on Webots for the simulation in Chapter 5.

8.3 . Relaxation of the Z -convex obstacle assumption

One of our strong assumptions is (3.1) which, according to (Zhang et al., 2023), considers that all the obstacles in an urban environment are Z -convex. This assumption strongly limits the possible application scenarios since tunnels, bridges, or buildings similar to the Pisa Tower may be encountered.

8.3.1 . Proposed approach

Assumption (3.1) has been made so that UAV i is able to process any *Obstacle*-labeled pixels to define an area in the neighborhood of the obstacle footprint, *i.e.*, $\mathbb{S}_m^o \cap \mathbb{X}_g$, that is guaranteed to not contain any target locations. Besides, we only consider ground targets. Consequently, they only evolve around the base of the obstacles. Thus, when UAV i characterizes $\mathbb{P}_{i,k}((n_r, n_c))$ for an *Obstacle*-labeled pixel, then UAV i can evaluate if (n_r, n_c) is looking at the base of the obstacle. Consequently, by defining the base of an obstacle as

$$\mathbb{S}_m^{\text{o,base}} = \mathbb{S}_m^o \cap \{ \mathbf{x} \in \mathbb{X}_0 \mid x_3 \in]-\infty, h^t] \}, \quad (8.2)$$

since a target is assumed to never collide with an obstacle, then if the UAV is able to prove that an Obstacle-labeled pixel is related to the base of an obstacle, this pixel can be used to characterize of $\underline{\mathbb{X}}_{i,k}^o$. This approach allows the presence of T-shape or V-shape obstacles within the RoI since the 'roof' of those obstacles is not taken into account, see Figure 8.3.

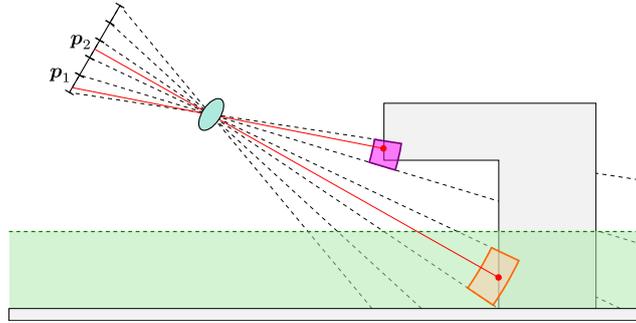


Figure 8.3 - Illustration of a T-shape obstacle observed by a UAV. The green area is $\{\mathbf{x} \in \mathbb{X}_0 \mid x_3 \in [0, h^t]\}$. Pixels \mathbf{p}_1 and \mathbf{p}_2 are labeled *Obstacle*. \mathbf{p}_1 cannot be related to the base of the obstacle because $\mathbb{P}_{i,k}(\mathbf{p}_1)$ (purple set) does not intersect with $\mathbb{S}_m^{o,base}$. \mathbf{p}_2 can be related to the base of an obstacle because $\mathbb{P}_{i,k}(\mathbf{p}_2)$ (orange set) is included in the green area.

Chapter A - Appendix

A.1 . Alternative target location estimator

This appendix considers how to design the target location estimator presented in Chapter 4 when either Assumption 5 or Assumption 6 is considered instead of Assumption 4. Since the only modification to the estimation process is the change of the assumption used to model how a pixel label can be related to the environment, only the approaches to characterize the sets $\mathbb{X}_{i,j,k}^{t,m}$, $\mathbb{X}_{i,k}^o$, and $\mathbb{X}_{i,j,k}^{t,m}$ are adapted. The prediction-correction process is unchanged. To lighten the notations in this appendix, the time index is omitted.

A.1.1 . Target localization - Assumption 2

Consider that Assumption 5 is used for *Target*-labeled pixels. This assumption states that if a pixel (n_r, n_c) is labeled *Target*, then there exists at least one light ray $\mathbf{v} \in \mathcal{V}_i^t(n_r, n_c)$ that illuminates the pixel (n_r, n_c) while stemming from a detected target. Nonetheless, $\mathcal{V}_i^t(n_r, n_c) \subset \mathcal{V}_i(n_r, n_c)$ is unknown. Consequently, only $\mathcal{V}_i(n_r, n_c)$ can be considered and, thus, the definition of $\mathbb{P}_i((n_r, n_c))$ introduced in (4.1) is unchanged.

Nonetheless, the proof of Proposition 9 must be adapted.

Proposition 24. *If $(n_r, n_c) \in \mathcal{Y}_i^t$, then there exists $\ell \in \mathcal{N}^t$ such that $\mathbb{P}_i((n_r, n_c)) \cap \mathbb{S}^t(\mathbf{x}_\ell^t) \neq \emptyset$.*

Démonstration. If $(n_r, n_c) \in \mathcal{Y}_i^t$, then according to Assumption 5, there exists a target $\ell \in \mathcal{N}^t$ such that there exists $\mathbf{v} \in \mathcal{V}_i^t(n_r, n_c) \subset \mathcal{V}_i(n_r, n_c)$ such that $\exists \mathbf{x}^* \in \mathbb{S}^t(\mathbf{x}_\ell^t) \subset \mathbb{X}_0$ such that $d_{\mathbf{v}}(\mathbf{x}_i^c, \{\mathbf{x}^*\}) = \mathbf{D}_i^0(n_r, n_c)$. This leads to have $d_{\mathbf{v}}(\mathbf{x}_i^c, \{\mathbf{x}^*\}) \in [\mathbf{D}_i](n_r, n_c)$. In addition, as $(n_r, n_c) \in \mathcal{Y}_i$, then we have $\frac{1}{1+w} \mathbf{D}_i(n_r, n_c) \leq d_{\max}$ and $d_{\mathbf{v}}(\mathbf{x}_i^c, \{\mathbf{x}^*\}) \leq d_{\max}$ according (3.36). Therefore, $\mathbf{x}^* \in \mathbb{F}(\mathbf{x}_i^u)$. Finally, from (4.1), we have $\mathbf{x}^* \in \mathbb{P}_i((n_r, n_c))$. So $\mathbb{P}_i((n_r, n_c)) \cap \mathbb{S}^t(\mathbf{x}_\ell^t) \neq \emptyset$. \square

The strong hypothesis in Assumption 5 is that the light ray \mathbf{v} that is used to define $\mathbf{D}_i^0(n_r, n_c)$ is constrained to belong to $\mathcal{V}_i^t(n_r, n_c)$ (assuming that the pixel (n_r, n_c) is labeled *Target*). If we do not have this strong hypothesis, then we might be in the case where the depth measurement is related to an entity (ground, obstacle,...) behind the detected target. See Figure A.1 as an illustration of this issue.

Then, Proposition 10, Corollary 11 and Proposition 13 are unchanged.

A.1.2 . Target localization - Assumption 3

Here, Assumption 6 is considered for the *Target*-labeled pixels. This assumption states that all the information associated to a pixel (n_r, n_c) can be related to the information that is carried by the light ray illuminating the center of (n_r, n_c) , i.e., $\mathbf{v}^c(n_r, n_c)$. Compared to Assumption 4 and 5, Assumption 6 clarifies which direction $\mathbf{v} \in \mathcal{V}_i(n_r, n_c)$ should be used to describe how a *Target*-labeled pixel can be related to the environment. Consequently, the definition of $\mathbb{P}_i((n_r, n_c))$ can be adapted into

$$\mathbb{P}_i((n_r, n_c)) = \{ \mathbf{x} \in \mathbb{F}(\mathbf{x}_i^u) \cap \mathbb{X}_0 \mid d_{\mathbf{v}^c(n_r, n_c)}(\mathbf{x}_i^c, \{\mathbf{x}\}) \in [\mathbf{D}_i](n_r, n_c) \}. \quad (\text{A.1})$$

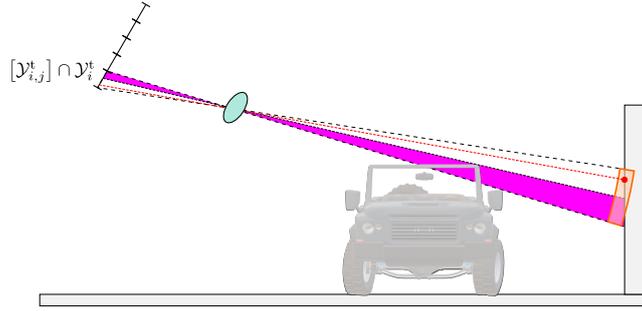


Figure A.1 – Illustration of an issue with $\mathbb{P}_i((n_r, n_c))$ (in orange) for a *target*-labeled pixel $(n_r, n_c) \in \mathcal{Y}_i^t \cap [\mathcal{Y}_{i,j}^t]$ while considering Assumption 5 but without assuming that the vector v such that $\mathbf{D}_i^0(n_r, n_c) = \rho(\mathbf{x}_i^c, v)$ belongs to $\mathcal{Y}_i^t(n_r, n_c)$ (the purple set). Without that assumption, even though the pixel is labeled *Target*, $\mathbb{P}_i((n_r, n_c))$ does not intersect with the shape of a target.

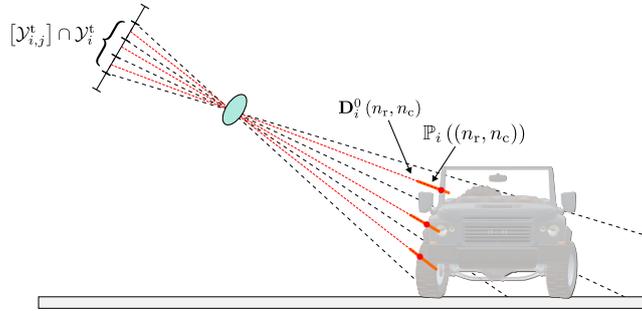


Figure A.2 – Illustration of the sets $\mathbb{P}_i((n_r, n_c))$ (in orange) for different *target*-labeled pixels $(n_r, n_c) \in \mathcal{Y}_i^t \cap [\mathcal{Y}_{i,j}^t]$ while considering Assumption 6

Figure A.2 illustrated this new definition of $\mathbb{P}_i((n_r, n_c))$ for three pixels labeled *Target*. Then, the proof of Proposition 9 must be adapted

Proposition 25. *If $(n_r, n_c) \in \mathcal{Y}_i^t$, then there exists $\ell \in \mathcal{N}^t$ such that $\mathbb{P}_i((n_r, n_c)) \cap \mathbb{S}^t(\mathbf{x}_\ell^t) \neq \emptyset$.*

Démonstration. If $(n_r, n_c) \in \mathcal{Y}_i^t$, then according to Assumption 6, there exists a target $\ell \in \mathcal{N}^t$ such that there exists a point $\mathbf{x}^* \in \mathbb{S}^t(\mathbf{x}_\ell^t) \subset \mathbb{X}_0$, such that $d_{v^c(n_r, n_c)}(\mathbf{x}_i^c, \{\mathbf{x}^*\}) = \mathbf{D}_i^0(n_r, n_c)$. This leads to have $d_{v^c(n_r, n_c)}(\mathbf{x}_i^c, \{\mathbf{x}^*\}) \in [\mathbf{D}_i](n_r, n_c)$. In addition, as $(n_r, n_c) \in \mathcal{Y}_i^t \subset \mathcal{Y}_i$, then we have $\frac{1}{1+w} \mathbf{D}_i(n_r, n_c) \leq d_{\max}$ and $d_v(\mathbf{x}_i^c, \{\mathbf{x}^*\}) \leq d_{\max}$ according (3.36). Therefore $\mathbf{x}^* \in \underline{\mathbb{F}}(\mathbf{x}_i^u)$. Finally, from (4.1), we have $\mathbf{x}^* \in \mathbb{P}_i((n_r, n_c))$. So $\mathbb{P}_i((n_r, n_c)) \cap \mathbb{S}^t(\mathbf{x}_\ell^t) \neq \emptyset$. \square

Then, Proposition 10, Corollary 11 and Proposition 13 are unchanged.

A.1.3 . Exploiting CVS information to characterize a target-free area

In addition of the set estimate $\underline{\mathbb{X}}_{i,j}^{t,m}$, with $j \in \mathcal{D}_i^t$, Chapter 4 introduced 3 more set estimates. The set estimate $\underline{\mathbb{P}}_i^g(\mathcal{Y}_i^g)$ is characterized from *Ground*-labeled pixels in order to get an area that cannot contain any target location. The set estimate $\underline{\mathbb{X}}_i^o$ is another set that cannot contain any target location due to the presence of an obstacle, and is characterized using *Obstacle*-labeled pixels. Lastly, the set

estimate $\underline{\mathbb{X}}_{i,j}^{t,m}$ represents an area that cannot contain the location of any target except the location of target j .

According to 3.54, the set estimate $\mathbb{P}_i^{\mathcal{E}}(\mathcal{Y}_i^{\mathcal{E}})$ cannot be characterized with Assumption 5 or Assumption 6. Then, the characterization of $\underline{\mathbb{X}}_i^0$ relies on the characterization of the set $\mathbb{P}_i((n_r, n_c))$ for pixels labeled *Obstacle*. If Assumption 5 is used for them, then nothing changes. If, instead, Assumption 6 is considered, then the characterization of $\underline{\mathbb{S}}((n_r, n_c), r_s^{t0})$ is simplified since $\mathbb{P}_i((n_r, n_c))$ will be a simple segment instead of a set. Lastly, the characterization of $\underline{\mathbb{X}}_{i,j}^{t,m}$ relies on $\underline{\mathbb{X}}_{i,j}^{t,m}$. Thus, nothing changes as well.

A.1.4 . Simulation

To compare Assumption 4, Assumption 5, and Assumption 6, we consider the simulation conditions as in Section 4.7.2 of Chapter 4. Figure A.3 illustrates the performance of the target location estimator for each of these assumptions while considering the metrics ϕ^{eR} and ϕ^c . Note that since $\mathbb{P}_i((n_r, n_c))$ is the same for Assumption 4 and Assumption 5, those two assumptions are plotted together in blue. The performance related to Assumption 6 is plotted in green.

Even though the target location estimation uncertainty, *i.e.*, $\phi^{eR}(\underline{\mathbb{X}}_{i,j,k}^{t,m})$, is lower with Assumption 6, $\phi^{eR}(\underline{\mathbb{X}}_{i,j,k}^t)$ and $\phi^c(\underline{\mathbb{X}}_{i,j,k}^t)$ are very similar. There is no real advantage to consider Assumption 6 instead of Assumption 4 or Assumption 5.

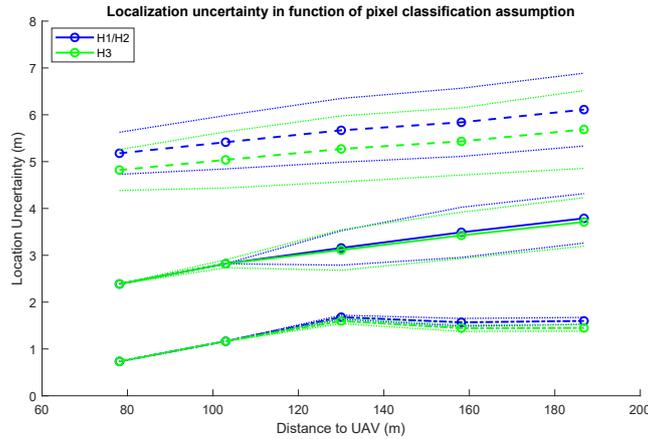


Figure A.3 – Mean value with standard deviation over 100 simulations of $\phi^{eR}(\underline{\mathbb{X}}_{i,j,k}^{t,m})$ (dashed lines - curves at the top of the figure), $\phi^{eR}(\underline{\mathbb{X}}_{i,j,k}^t)$ (solid lines - curves in the middle of the figure), and $\phi^c(\underline{\mathbb{X}}_{i,j,k}^t)$ (broken lines - curves at the bottom of the figure). The green curves are for Assumption 6. The blue curves are for Assumption 4 and 5.

A.2 . Practical implementation

This section describes an approach to get an outer-approximation of the set estimate $\underline{\mathbb{X}}_{i,j,k}^{t,m}$ introduced in Section 4.2, an inner-approximation of $\underline{\mathbb{X}}_{i,k}^0$ and $\underline{\mathbb{X}}_{i,j,k}^{t,m}$ introduced in Section 4.3.2 and 4.3.3,

an inner-approximation of $\mathbb{P}_{i,k}^g \left(\mathcal{Y}_{i,k}^g \right)$ introduced in Section 4.3.1, and an inner-approximation of $\mathbb{H}_{i,k}^{\text{CVS}}$ introduced in Section 4.4.

The determination of $\mathbb{X}_{i,j,k}^{\text{t,m}}$, $\mathbb{X}_{i,k}^{\text{o}}$ and $\mathbb{X}_{i,j,k}^{\text{t,m}}$ require the characterization of $\mathbb{P}_{i,k}((n_r, n_c))$ and $\mathbb{S}((n_r, n_c), r)$. Thus, their characterization is explained as well.

To lighten the notations, the time index is omitted.

The operations (union, intersection, ...) between 2D sets described as polygons are performed using the polshape library of MATLAB.

A.2.1 . Characterization of $\mathbb{P}_{i,k}((n_r, n_c))$

Consider a pixel (n_r, n_c) . The set $\mathbb{P}_i((n_r, n_c))$, as defined in (4.1), is the intersection of *i*) the set of points between the sphere of center \mathbf{x}_i^c and radius $\frac{1}{1+w} \mathbf{D}_i(n_r, n_c)$ and the sphere of center \mathbf{x}_i^c and radius $\frac{1}{1+w} \mathbf{D}_i(n_r, n_c)$ and *ii*) the half cone with apex \mathbf{x}_i^c and edges $\mathbf{M}_{\mathcal{F}_i^c}^{\mathcal{F}_i^c} \mathbf{v}_{\ell,r,c}^{\mathcal{F}_i^c}$ with $\ell = 1, \dots, 4$. The vectors $\mathbf{v}_{\ell,r,c}^{\mathcal{F}_i^c} = \mathbf{v}^{\mathcal{F}_i^c}(x_{\ell,r,c}, y_{\ell,r,c})$ for $\ell = 1, \dots, 4$, with

$$(x_{\ell,r,c}, y_{\ell,r,c}) \in \left\{ \begin{array}{ll} (n_c, n_r - 1), & (n_c - 1, n_r - 1), \\ (n_c, n_r), & (n_c - 1, n_r) \end{array} \right\}.$$

See Figure 3.7 where the 4 vectors $\mathbf{v}_{\ell}^{\mathcal{F}_i^c}$ are represented by the blue lines.

A truncated pyramidal outer-approximation $\overline{\mathbb{P}}_i((n_r, n_c))$ of $\mathbb{P}_i((n_r, n_c))$ is easily built. For that purpose, consider the angle θ_ℓ between the $\mathbf{v}_{\ell,r,c}^{\mathcal{F}_i^c}$ with $\mathbf{v}^{c,\mathcal{F}_i^c}(n_c, n_r)$, the unit vector defining the light-ray illuminating the center of the pixel (n_r, n_c) . The eight vertices defining $\overline{\mathbb{P}}_i((n_r, n_c))$ are

$$\underline{\mathbf{x}}_\ell^p = \mathbf{x}_i^c + \frac{1}{1+w} \mathbf{D}_i(n_r, n_c) \mathbf{M}_{\mathcal{F}_i^c}^{\mathcal{F}_i^c} \mathbf{v}_{\ell,r,c}^{\mathcal{F}_i^c}, \quad (\text{A.2})$$

$$\overline{\mathbf{x}}_\ell^p = \mathbf{x}_i^c + \frac{1}{\cos(\theta_\ell)} \frac{1}{1+w} \mathbf{D}_i(n_r, n_c) \mathbf{M}_{\mathcal{F}_i^c}^{\mathcal{F}_i^c} \mathbf{v}_{\ell,r,c}^{\mathcal{F}_i^c}, \quad (\text{A.3})$$

with $\ell \in \{1, 2, 3, 4\}$. The set $\overline{\mathcal{P}}_i(n_r, n_c) = \{\underline{\mathbf{x}}_1^p, \dots, \underline{\mathbf{x}}_4^p, \overline{\mathbf{x}}_1^p, \dots, \overline{\mathbf{x}}_4^p\}$ contains the vertices defining $\overline{\mathbb{P}}_i((n_r, n_c))$.

A.2.2 . Characterization of $\mathbb{X}_{i,j,k}^{\text{t,m}}$

The evaluation of $\mathbb{X}_{i,j}^{\text{t,m}}$ requires the determination of the projection $\mathbf{p}_g \left(\mathbb{P}_{i,j}^{\text{t}} \right)$ of $\mathbb{P}_{i,j}^{\text{t}}$ on the ground \mathbb{X}_g , where $\mathbb{P}_{i,j}^{\text{t}}$ is the union of the sets $\mathbb{P}_i((n_r, n_c))$ over all $(n_r, n_c) \in \left[\mathcal{Y}_{i,j}^{\text{t}} \right] \cap \mathcal{Y}_i^{\text{t}}$, see Proposition 13. Therefore, $\mathbf{p}_g \left(\mathbb{P}_{i,j}^{\text{t}} \right)$ is the union of the sets $\mathbf{p}_g \left(\mathbb{P}_i((n_r, n_c)) \right)$ over $(n_r, n_c) \in \left[\mathcal{Y}_{i,j}^{\text{t}} \right] \cap \mathcal{Y}_i^{\text{t}}$. Since $\mathbb{P}_i((n_r, n_c)) \subset \overline{\mathbb{P}}_i((n_r, n_c))$ and $\overline{\mathbb{P}}_i((n_r, n_c))$ is a convex set, an outer-approximation of $\mathbf{p}_g \left(\mathbb{P}_i((n_r, n_c)) \right)$ is obtained by characterizing the convex hull of $\mathbf{p}_g \left(\overline{\mathbb{P}}_i((n_r, n_c)) \right)$. Finally, an outer-approximation of $\mathbb{X}_{i,j}^{\text{t,m}}$ is obtained as $\mathbb{X}_g \cap \left(\mathbf{p}_g \left(\overline{\mathbb{P}}_{i,j} \right) \oplus \mathbb{N}_g \left(\{ \mathbf{0} \}, r^{\text{t}} \right) \right)$, with $\mathbf{p}_g \left(\overline{\mathbb{P}}_{i,j} \right)$ being the union of $\mathbf{p}_g \left(\overline{\mathbb{P}}_i((n_r, n_c)) \right)$ over $(n_r, n_c) \in \left[\mathcal{Y}_{i,j}^{\text{t}} \right] \cap \mathcal{Y}_i^{\text{t}}$.

Since $\mathbb{N}_g \left(\{ \mathbf{0} \}, r^{\text{t}} \right)$ is just a disc of radius r^{t} and centered at the origin, the Minkovski sum $\mathbf{p}_g \left(\overline{\mathbb{P}}_{i,j} \right) \oplus \mathbb{N}_g \left(\{ \mathbf{0} \}, r^{\text{t}} \right)$ is a dilatation of $\mathbf{p}_g \left(\overline{\mathbb{P}}_{i,j} \right)$ by a distance r^{t} , *i.e.*,

$$\mathbb{X}_g \cap \left(\mathbf{p}_g \left(\overline{\mathbb{P}}_{i,j} \right) \oplus \mathbb{N}_g \left(\{ \mathbf{0} \}, r^{\text{t}} \right) \right) = \mathbb{N}_g \left(\mathbf{p}_g \left(\overline{\mathbb{P}}_{i,j} \right), r^{\text{t}} \right).$$

A.2.3 . Characterization of $\underline{\mathbb{S}}((n_r, n_c), r_s^{\text{to}})$

According to (4.7), $\underline{\mathbb{S}}((n_r, n_c), r_s^{\text{to}})$ is the intersection of all discs $\mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{to}})$ of radius r_s^{to} and of center \mathbf{x} taken in $\mathbf{p}_g(\mathbb{P}_i((n_r, n_c)))$. Since $\mathbb{P}_i((n_r, n_c)) \subset \overline{\mathbb{P}}_i((n_r, n_c))$, one has $\mathbf{p}_g(\mathbb{P}_i((n_r, n_c))) \subset \mathbf{p}_g(\overline{\mathbb{P}}_i((n_r, n_c)))$. Consequently

$$\bigcap_{\mathbf{x} \in \mathbf{p}_g(\overline{\mathbb{P}}_i((n_r, n_c)))} \mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{to}}) \subset \bigcap_{\mathbf{x} \in \mathbf{p}_g(\mathbb{P}_i((n_r, n_c)))} \mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{to}}). \quad (\text{A.4})$$

Therefore, we have

$$\bigcap_{\mathbf{x} \in \mathbf{p}_g(\overline{\mathbb{P}}_i((n_r, n_c)))} \mathbb{N}(\{\mathbf{x}\}, r_s^{\text{to}}) \subset \underline{\mathbb{S}}((n_r, n_c), r_s^{\text{to}}). \quad (\text{A.5})$$

Moreover, as $\overline{\mathbb{P}}_i((n_r, n_c))$ is convex, we have

$$\bigcap_{\mathbf{x} \in \mathbf{p}_g(\overline{\mathbb{P}}_i((n_r, n_c)))} \mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{to}}) = \bigcap_{\mathbf{x} \in \mathbf{p}_g(\overline{\mathcal{P}}_i(n_r, n_c))} \mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{to}}), \quad (\text{A.6})$$

An inner approximation of $\underline{\mathbb{S}}((n_r, n_c), r_s^{\text{to}})$ is then obtained as the intersection of, at most, eight discs each with a center in the set $\mathbf{p}_g(\overline{\mathcal{P}}_i(n_r, n_c))$ and of radius r_s^{to} . This inner approximation can be computed with polyshape by approximating each disc $\mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{to}})$, with $\mathbf{x} \in \mathbf{p}_g(\overline{\mathcal{P}}_i(n_r, n_c))$, by a polygon contained in $\mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{to}})$ where each of its vertices is a point belonging to the circle associated to $\mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{to}})$.

A.2.4 . Characterization of $\underline{\mathbb{X}}_{i,k}^{\text{o}}$ and $\underline{\mathbb{X}}_{i,j}^{\text{t,m}}$

The set $\underline{\mathbb{X}}_i^{\text{o}}$, introduced in Proposition 17, is the union of the sets $\underline{\mathbb{S}}((n_r, n_c), r_s^{\text{to}})$, for all $(n_r, n_c) \in \mathcal{Y}_i^{\text{o}}$.

The characterization of an inner-approximation of $\underline{\mathbb{X}}_{i,j}^{\text{t,m}}$ can be done by characterizing the convex hull $\langle \underline{\mathbb{X}}_{i,j}^{\text{t,m}} \rangle$ of $\underline{\mathbb{X}}_{i,j}^{\text{t,m}}$. Since $\underline{\mathbb{X}}_{i,j}^{\text{t,m}} \subset \langle \underline{\mathbb{X}}_{i,j}^{\text{t,m}} \rangle$, then

$$\bigcap_{\mathbf{x} \in \langle \underline{\mathbb{X}}_{i,j}^{\text{t,m}} \rangle} \mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{tt}}) \subset \bigcap_{\mathbf{x} \in \underline{\mathbb{X}}_{i,j}^{\text{t,m}}} \mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{tt}}). \quad (\text{A.7})$$

Therefore, we have

$$\bigcap_{\mathbf{x} \in \langle \underline{\mathbb{X}}_{i,j}^{\text{t,m}} \rangle} \mathbb{N}_g(\{\mathbf{x}\}, r_s^{\text{tt}}) \subset \underline{\mathbb{X}}_{i,j}^{\text{t,m}}. \quad (\text{A.8})$$

Moreover, as $\langle \underline{\mathbb{X}}_{i,j}^{\text{t,m}} \rangle$ is a convex polygon, instead of evaluating the intersection of all the discs of radius r_s^{tt} which have their center taken in $\langle \underline{\mathbb{X}}_{i,j}^{\text{t,m}} \rangle$, we rather compute the intersection of the discs of radius r_s^{tt} which have their center taken as the vertices of $\langle \underline{\mathbb{X}}_{i,j}^{\text{t,m}} \rangle$. The same trick used to characterize an inner approximation of $\underline{\mathbb{S}}((n_r, n_c), r_s^{\text{to}})$ is used here.

A.2.5 . Characterization of $\mathbb{P}_{i,k}^{\mathcal{G}}(\mathcal{Y}_{i,k}^{\mathcal{G}})$ and $\mathbb{H}_{i,k}^{\text{CVS}}$

The set $\mathbb{P}_i^{\mathcal{G}}(\mathcal{Y}_i^{\mathcal{G}})$, introduced in (4.6), is the union of sets $\mathbb{P}_i^{\mathcal{G}}((n_r, n_c))$ for $(n_r, n_c) \in \mathcal{Y}_i^{\mathcal{G}}$. Neglecting here the limitation of the FoV by $\mathbb{B}(\mathbf{x}_i^u, d_{\max})$ and as \mathbb{X}_g is a known horizontal plane, $\mathbb{P}_i^{\mathcal{G}}((n_r, n_c))$ is the convex quadrangle defined by the intersection of \mathbb{X}_g with the half-cone of apex \mathbf{x}_i^c and edges related to the four unit vector $\mathbf{v}_{\ell,r,c}^{\mathcal{F}_i^c}$ with $\ell = 1, \dots, 4$, see Figure 3.7.

$\mathbb{P}_i^{\mathcal{G}}(\mathcal{Y}_i^{\mathcal{G}})$ is then the union of the convex quadrangles $\mathbb{P}_i^{\mathcal{G}}((n_r, n_c))$ for all $(n_r, n_c) \in \mathcal{Y}_i^{\mathcal{G}}$. Similarly, $\mathbb{H}_i^{\text{CVS}}$ is the union of the convex quadrangles $\mathbb{P}_i^{\mathcal{G}}((n_r, n_c))$ for all $(n_r, n_c) \in \mathcal{Y}_i^o \cup \mathcal{Y}_i^t \cup \mathcal{Y}_i^n$.

To account for $\mathbb{B}(\mathbf{x}_i^u, d_{\max})$, one can evaluate the disc defined by the intersection $\mathbb{B}(\mathbf{x}_i^u, d_{\max}) \cap \{\mathbf{x} \in \mathbb{R}^3 \mid x_3 = 0\}$, and then intersect the resulting disc with $\mathbb{P}_i^{\mathcal{G}}(\mathcal{Y}_i^{\mathcal{G}})$ or $\mathbb{H}_i^{\text{CVS}}$.

A.3 . Interval Analysis and SIVIA

The appendix presents what is interval analysis and the Set Inverter Via Interval Analysis (SIVIA). Most of what is presented in this appendix comes from (Jaulin et al., 2001).

A.3.1 . Preliminaries

This section introduces interval arithmetic while recalling some results and properties.

An interval $[x] = [\underline{x}, \bar{x}]$ of real numbers is defined as the subset of \mathbb{R} such that

$$[x] = \{x \in \mathbb{R} \mid \underline{x} \leq x \text{ and } x \leq \bar{x}\}, \quad (\text{A.9})$$

where \underline{x} and \bar{x} are respectively the lower and upper bounds of the interval. Another definition exists which does not rely on the bounds \underline{x} and \bar{x} of the interval but rather on its center $\frac{\bar{x} + \underline{x}}{2}$ and radius $\frac{\bar{x} - \underline{x}}{2}$.

The set containing all the real intervals is denoted \mathbb{IR} . A box $[\mathbf{x}] \in \mathbb{IR}^n$ is a vector of n components where all of its components are real intervals. In Appendix A.4, the notion of thick interval (and thus thick boxes) is used. A thick interval $[[x]] = [[x^-], [x^+]] \in \mathbb{IR}$ is a set of intervals $[x] = [\underline{x}, \bar{x}] \in \mathbb{IR}$ where their lower and upper bounds have to be contained in an interval of lower bounds $[x^-]$ and an interval of upper bounds $[x^+]$, *i.e.*,

$$[[x]] = \{[x] = [\underline{x}, \bar{x}] \in \mathbb{IR} \mid \underline{x} \in [x^-] \text{ and } \bar{x} \in [x^+]\}. \quad (\text{A.10})$$

Let consider a measurement function $\mathbf{h} : \mathbb{X}_0 \times \mathbb{R}^n \rightarrow \mathbb{R}^m$ such that

$$\mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{p}) + \mathbf{w}, \quad (\text{A.11})$$

where $\mathbf{y} \in \mathbb{R}^m$ is the measurement of $\mathbf{x} \in \mathbb{X}_0$, \mathbf{p} is an unknown vector of parameters of \mathbf{h} , and $\mathbf{w} \in \mathbb{R}^m$ is the unknown but bounded measurement noise contained in $[\mathbf{w}] \in \mathbb{IR}^m$. The vector of parameters is assumed bounded and contained in the known box $[\mathbf{p}]$. One may deduce that the interval $[\mathbf{y}] = \{\mathbf{y}\} - [\mathbf{w}]$ contains the noise-free measurement, *i.e.*, $\mathbf{h}(\mathbf{x}, \mathbf{p}) \in [\mathbf{y}]$.

Assuming that \mathbf{x} belongs to the shape of an object $\mathbb{S} \subset \mathbb{X}_0$, then, one may estimate the location of that object from the noisy measurement $[\mathbf{y}]$. Since the vector of parameters \mathbf{p} is uncertain, there exists two solution sets that result from this localization problem. Either one wants to characterize a set of locations that are consistent with *all* possible values of the vector of parameters :

$$\mathbb{X}^{\forall} = \{\mathbf{x} \in \mathbb{X}_0 \mid \forall \mathbf{p} \in [\mathbf{p}], \mathbf{x} = \mathbf{h}^{-1}([\mathbf{y}], \mathbf{p})\}, \quad (\text{A.12})$$

or one wants to get the set of locations that are consistent with *at least one* value $\mathbf{p} \in [\mathbf{p}]$, *i.e.*,

$$\mathbb{X}^{\exists} = \{ \mathbf{x} \in \mathbb{X}_0 \mid \exists \mathbf{p} \in [\mathbf{p}], \mathbf{x} = \mathbf{h}^{-1}([\mathbf{y}], \mathbf{p}) \}. \quad (\text{A.13})$$

Nonetheless, the function \mathbf{h} could be non-linear and, thus, hard to inverse. This is why one prefers to solve the set inversion problem defined by

$$\mathbb{X}^{\forall} = \{ \mathbf{x} \in \mathbb{X}_0 \mid \forall \mathbf{p} \in [\mathbf{p}], \mathbf{h}(\mathbf{x}, \mathbf{p}) \in [\mathbf{y}] \}, \quad (\text{A.14})$$

$$\mathbb{X}^{\exists} = \{ \mathbf{x} \in \mathbb{X}_0 \mid \exists \mathbf{p} \in [\mathbf{p}], \mathbf{h}(\mathbf{x}, \mathbf{p}) \in [\mathbf{y}] \}. \quad (\text{A.15})$$

Figure A.4 illustrates \mathbb{X}^{\forall} in green and \mathbb{X}^{\exists} in blue (and green). The black sets correspond to

$$\mathbb{X}(\mathbf{p}) = \{ \mathbf{x} \in \mathbb{X}_0 \mid \mathbf{h}(\mathbf{x}, \mathbf{p}) \in [\mathbf{y}] \} \quad (\text{A.16})$$

for three different values of $\mathbf{p} \in [\mathbf{p}]$. Alternative formulations of A.14 and A.15 are

$$\mathbb{X}^{\forall} = \bigcap_{\mathbf{p} \in [\mathbf{p}]} \mathbb{X}(\mathbf{p}), \quad (\text{A.17})$$

$$\mathbb{X}^{\exists} = \bigcup_{\mathbf{p} \in [\mathbf{p}]} \mathbb{X}(\mathbf{p}). \quad (\text{A.18})$$

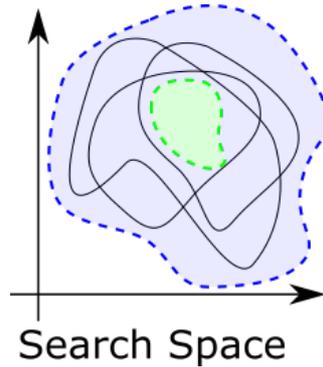


Figure A.4 – Illustration of \mathbb{X}^{\forall} and \mathbb{X}^{\exists} . The black sets are potential solution sets for a specific value of the vector of parameters.

A.3.2 . Inclusion function

The image of \mathbf{h} for some $\mathbf{x} \in \mathbb{X}_0$ and some $\mathbf{p} \in [\mathbf{p}]$ is defined as $\mathbf{h}(\mathbf{x}, \mathbf{p})$. Thus, the range of \mathbf{h} over $[\mathbf{x}] \times [\mathbf{p}]$ is defined as the set containing all the images of \mathbf{h} for any input $(\mathbf{x}, \mathbf{p}) \in [\mathbf{x}] \times [\mathbf{p}]$, with $[\mathbf{x}] \subset \mathbb{X}_0$. The expression of this set is

$$\mathbf{h}([\mathbf{x}], [\mathbf{p}]) = \{ \mathbf{h}(\mathbf{x}, \mathbf{p}) \mid \mathbf{x} \in [\mathbf{x}], \mathbf{p} \in [\mathbf{p}] \}. \quad (\text{A.19})$$

Nonetheless, $\mathbf{h}([\mathbf{x}], [\mathbf{p}])$ may be hard or impossible to evaluate due to non-linearities. To approximate $\mathbf{h}([\mathbf{x}], [\mathbf{p}])$, one may use interval analysis to get an inclusion function $[\mathbf{h}]$ of \mathbf{h} . By definition, an inclusion function $[\mathbf{h}]$ of \mathbf{h} provides a box $[\mathbf{h}]([\mathbf{x}], [\mathbf{p}])$ that contains $\mathbf{h}([\mathbf{x}], [\mathbf{p}])$, *i.e.*,

$$\mathbf{h}([\mathbf{x}], [\mathbf{p}]) \subset [\mathbf{h}]([\mathbf{x}], [\mathbf{p}]). \quad (\text{A.20})$$

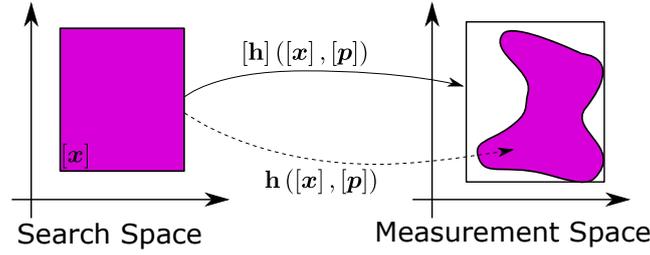


Figure A.5 – Illustration of an inclusion function and the pessimism induced by this approximation.

Figure A.5 illustrates a possible box that may be obtained for a possible inclusion function $[h]$ of h .

Natural inclusion functions are obtained by replacing arithmetic operations and elementary functions with their interval counterpart, see (Jaulin et al., 2001) for more details.

Using inclusion functions, several properties can be obtained.

Proposition 26. *If $[h]([x], [p]) \subset [y]$, then $[x] \subset \mathbb{X}^\forall$*

Démonstration. If $[h]([x], [p]) \subset [y]$, then, according to (A.20), one has $h([x], [p]) \subset [y]$. Therefore, for all $x \in [x]$ and for all $p \in [p]$, $h(x, p) \in [y]$. Consequently, $[x] \subset \mathbb{X}^\forall$. \square

Proposition 27. *If $[h]([x], [p]) \cap [y] = \emptyset$, then $[x] \cap \mathbb{X}^\forall = \emptyset$*

Démonstration. If $[h]([x], [p]) \cap [y] = \emptyset$, then, according to (A.20), one has $h([x], [p]) \cap [y] = \emptyset$. Therefore, for all $x \in [x]$ and for all $p \in [p]$, $h(x, p) \notin [y]$. Consequently, $[x] \cap \mathbb{X}^\forall = \emptyset$. \square

A.3.3 . Set Inverter Via Interval Analysis

The Set Inverter Via Interval Analysis (SIVIA) (Jaulin and Walter, 1993) can be used to characterize \mathbb{X}^\forall or \mathbb{X}^\exists , assuming that an inclusion function $[h]$ of h is available. This algorithm characterizes an inner and an outer-approximation of \mathbb{X}^\forall by building iteratively a list of non-overlapping boxes. SIVIA relies on the properties 26 and 27. The first one allows to test if the image of $[x]$ by $[h]$ while knowing $[p]$ is included inside $[y]$ to prove that $[x] \subset \mathbb{X}^\forall$, i.e., $[x]$ only contains solutions of the inverse problem related to \mathbb{X}^\forall . The second allows to test if the image of $[x]$ by $[h]$ while knowing $[p]$ has no intersection with $[y]$ to prove that $[x] \cap \mathbb{X}^\forall = \emptyset$, i.e., $[x]$ does not contain any solution. If none of those tests is satisfied, then $[x]$ is proved to contain at least one solution x_1 such that $x_1 \in \mathbb{X}^\forall$ while containing at least a x_2 such that $x_2 \notin \mathbb{X}^\forall$. Therefore, $[x]$ is bisected, and the resulting boxes, i.e., its children, are tested with the same process. This scheme continues until the tested box is considered too small regarding some threshold ε to be bisected again.

Starting with an initial box contained in \mathbb{X}_0 , SIVIA provides

- a subpaving of boxes \mathcal{B}_{in} such that all $[x] \in \mathcal{B}_{in}$ satisfy Proposition 26
- a subpaving of boxes \mathcal{B}_{out} such that all $[x] \in \mathcal{B}_{out}$ satisfy Proposition 27
- a subpaving of boxes \mathcal{B}_{border} such that none $[x] \in \mathcal{B}_{border}$ satisfy Propositions 26 and 27 but are deemed too small to be bisected again.

Consequently, using SIVIA and a starting box sufficiently large to ensure that it contains \mathbb{X}^\exists , one is able to get an inner-approximation of \mathbb{X}^\forall with \mathcal{B}_{in} and an outer-approximation of \mathbb{X}^\exists with $\mathcal{B}_{in} \cup \mathcal{B}_{border}$.

$\mathcal{B}_{\text{border}}$ has to be considered for \mathbb{X}^{\exists} because each $[\mathbf{x}] \in \mathcal{B}_{\text{border}}$ contains solutions of the inverse problem (A.15).

Figure A.6 illustrates the application of SIVIA when one tries to characterize the set

$$\mathbb{X} = \left\{ \mathbf{x} \in [-5, 5]^2 \mid \mathbf{h}(\mathbf{x}, \mathbf{p}) \in [2, 4] \right\} \quad (\text{A.21})$$

where $\mathbf{p} = (0, 0)^T$ and $\mathbf{h}(\mathbf{x}, \mathbf{p}) = \|\mathbf{x} - \mathbf{p}\|$.

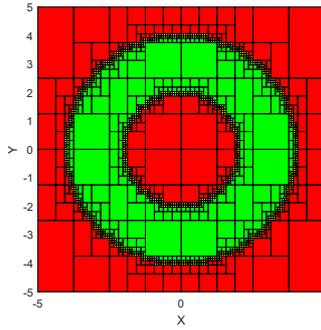


Figure A.6 – Characterization of the set defined by (A.21) using SIVIA. \mathcal{B}_{in} is the 304 green boxes. \mathcal{B}_{out} is the red boxes. $\mathcal{B}_{\text{border}}$ is the 920 yellow boxes.

Unfortunately, the capacity of SIVIA to provide an outer-approximation of \mathbb{X}^{\exists} is also what made the algorithm unfit to solve set inversion problems that involve an uncertain vector of parameters, see Figure A.7 where the vector of parameters \mathbf{p} is only known to belong to $[-0.5, 0.5]^2$.

Due to the recursive approach based on testing and bisecting, the termination of the algorithm only occurs when all the remaining boxes that do not satisfy 26 and 27 are too small to be bisected. Therefore, since $\mathcal{B}_{\text{border}}$ is also an outer-approximation of $\mathbb{X}^{\exists} \setminus \mathbb{X}^{\forall}$, if the uncertainty regarding the vector of parameters $[\mathbf{p}]$ becomes large, then it is likely that $\mathbb{X}^{\exists} \setminus \mathbb{X}^{\forall}$ grows. This leads to more boxes of the minimal size in $\mathcal{B}_{\text{border}}$ and thus a higher computational time to only obtain an inner-approximation of \mathbb{X}^{\forall} , *i.e.*, \mathcal{B}_{in} .

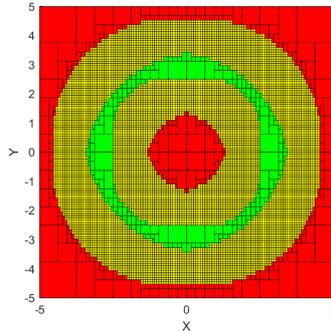


Figure A.7 – Characterization of the set defined by (A.21) using SIVIA while considering an uncertain vector of parameters. \mathcal{B}_{in} is the 236 green boxes. \mathcal{B}_{out} is the red boxes. $\mathcal{B}_{\text{border}}$ is the 8672 yellow boxes.

This problem of efficiency comes from the fact that inclusion functions are unable to determine whether $\forall x \in [x], \exists p \in [p]$ such that $\mathbf{h}(x, p) \in [y]$. Thick intervals and Thick-SIVIA address this issue, see (Desrochers and Jaulin, 2017).

A.4 . Thick intervals & Thick-SIVIA

This appendix presents an approach to improve the performance of SIVIA when one want to characterize as efficiently as possible sets that are defined by either (A.14) or (A.15). Most of what is presented in this appendix comes from (Desrochers and Jaulin, 2017).

A.4.1 . Set inversion reformulation

Consider the set of functions

$$\mathcal{H}([p]) = \{f \in \mathcal{F}(\mathbb{X}_0, \mathbb{R}^m) \mid \exists p \in [p], \forall x \in \mathbb{X}_0, f(x) = \mathbf{h}(x, p)\}$$

where $\mathcal{F}(\mathbb{X}_0, \mathbb{R}^m)$ is the set of functions going from \mathbb{X}_0 to \mathbb{R}^m . This set contains all the measurement functions for a specific value $p \in [p]$ of the vector of parameters. From $\mathcal{H}([p])$, the set \mathbb{X}^\exists A.15 may be redefined as

$$\mathbb{X}^\exists = \{x \in \mathbb{X}_0 \mid \exists f \in \mathcal{H}([p]), f(x) \in [y]\}, \quad (\text{A.22})$$

Similarly to (A.22), an alternative formulation of \mathbb{X}^\forall is

$$\mathbb{X}^\forall = \{x \in \mathbb{X}_0 \mid \forall f \in \mathcal{H}([p]), f(x) \in [y]\}. \quad (\text{A.23})$$

To efficiently characterize \mathbb{X}^\exists and \mathbb{X}^\forall , one has to resort to thick intervals and thick inclusion functions.

A.4.2 . Thick sets and thick inclusion function

This section introduces the thick function $\llbracket \mathbf{h} \rrbracket$ of a measurement function \mathbf{h} as well as its thick inclusion function $\llbracket \mathbf{h} \rrbracket$. Once introduced, several properties are recalled.

A thick function $\llbracket \mathbf{g} \rrbracket$ is a set of functions f for which their image $f(x)$, for an input x , always belong to a box $[g(x)]$ of known bounds. Thus, to obtain a thick function of \mathbf{h} , we define for all $x \in \mathbb{X}_0$

$$\mathbf{h}^-(x) = \min_{p \in [p]} \mathbf{h}(x, p), \quad (\text{A.24})$$

$$\mathbf{h}^+(x) = \max_{p \in [p]} \mathbf{h}(x, p), \quad (\text{A.25})$$

where the minimization and the maximization are performed component-wise. Naturally, one has

$$\forall p \in [p], \forall x \in \mathbb{X}_0, \mathbf{h}^-(x) \leq \mathbf{h}(x, p) \leq \mathbf{h}^+(x). \quad (\text{A.26})$$

From \mathbf{h}^- and \mathbf{h}^+ the thick function defined as

$$\llbracket \mathbf{h} \rrbracket = [\mathbf{h}^-, \mathbf{h}^+] \quad (\text{A.27})$$

$$= \{f \in \mathcal{F}(\mathbb{X}_0, \mathbb{R}^m) \mid \forall x \in \mathbb{X}_0, \mathbf{h}^-(x) \leq f(x) \leq \mathbf{h}^+(x)\} \quad (\text{A.28})$$

is such that one has

$$\forall \mathbf{f} \in \mathcal{H}([p]), \forall \mathbf{x} \in \mathbb{X}_0, \mathbf{h}^-(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}) \leq \mathbf{h}^+(\mathbf{x}). \quad (\text{A.29})$$

Consequently, the thick function $[\mathbf{h}]$ contains, among others, all the functions $\mathcal{H}([p])$, *i.e.*,

$$\mathcal{H}([p]) \subset [\mathbf{h}]. \quad (\text{A.30})$$

Consider now the set

$$\overline{\mathbb{X}}^\exists = \{\mathbf{x} \in \mathbb{X}_0 \mid \exists \mathbf{f} \in [\mathbf{h}], \mathbf{f}(\mathbf{x}) \in [\mathbf{y}]\}. \quad (\text{A.31})$$

From (A.22) and (A.30), one has $\overline{\mathbb{X}}^\exists \subset \overline{\mathbb{X}}^\exists$. Similarly, one has $\underline{\mathbb{X}}^\forall \subset \underline{\mathbb{X}}^\forall$ with

$$\underline{\mathbb{X}}^\forall = \{\mathbf{x} \in \mathbb{X}_0 \mid \forall \mathbf{f} \in [\mathbf{h}], \mathbf{f}(\mathbf{x}) \in [\mathbf{y}]\}. \quad (\text{A.32})$$

Consider two inclusion functions $[\mathbf{h}^-]$ and $[\mathbf{h}^+]$ of \mathbf{h}^- and \mathbf{h}^+ . A thick inclusion function (TIF) $[[\mathbf{h}]]$ associated to $[\mathbf{h}]$ is the set of all inclusion functions $[\mathbf{f}]$ such that for all $[x] \subset \mathbb{X}_0$, its image $[\mathbf{f}]([x]) = [\underline{\mathbf{y}}, \overline{\mathbf{y}}]$ satisfies

$$\underline{\mathbf{y}} \in [\mathbf{h}^-]([x]) \text{ and } \overline{\mathbf{y}} \in [\mathbf{h}^+]([x]). \quad (\text{A.33})$$

The TIF $[[\mathbf{h}]]$ may also be written as $[[\mathbf{h}]] = [[[\mathbf{h}^-], [\mathbf{h}^+]]]$, and the image of some $[x] \subset \mathbb{X}_0$ by the TIF $[[\mathbf{h}]]$ is the set of boxes $[\underline{\mathbf{y}}, \overline{\mathbf{y}}] \in \mathbb{I}\mathbb{R}^m$ such that $\underline{\mathbf{y}}$ belong to the box of lower bounds $[\mathbf{h}^-]([x])$ and $\overline{\mathbf{y}}$ belongs to the box of upper bounds $[\mathbf{h}^+]([x])$, *i.e.*,

$$[[\mathbf{h}]]([x]) = \{[\underline{\mathbf{y}}, \overline{\mathbf{y}}] \in \mathbb{I}\mathbb{R}^m \mid \underline{\mathbf{y}} \in [\mathbf{h}^-]([x]), \overline{\mathbf{y}} \in [\mathbf{h}^+]([x])\}. \quad (\text{A.34})$$

$[[\mathbf{h}]]([x])$ is a thick box. Figure A.8 shows an example of a thick box.

Example

Consider the function $f(x, p) = x - p$. The input x is uncertain but contained in $[x] = [0, 10]$. Similarly, one assumes to have $p \in [p] = [0, 15]$.

Basic interval analysis gives

$$[f]([x], [p]) = [x] - [p] \quad (\text{A.35})$$

$$= [\underline{x} - \overline{p}, \overline{x} - \underline{p}] \quad (\text{A.36})$$

$$= [-15, 10] \quad (\text{A.37})$$

If one computes the thick function of f , one obtains

$$f^+(x) = \max_{p \in [p]} f(x, p) \quad (\text{A.38})$$

$$= x - \underline{p} \quad (\text{A.39})$$

$$f^-(x) = \min_{p \in [p]} f(x, p) \quad (\text{A.40})$$

$$= x - \overline{p} \quad (\text{A.41})$$

Then, we deduce a TIF of f

$$\llbracket f \rrbracket ([x]) = \llbracket [f^-([x]), [f^+([x])] \rrbracket, \quad (\text{A.42})$$

$$= \llbracket [x] - 15, [x] - 0 \rrbracket, \quad (\text{A.43})$$

$$\llbracket f \rrbracket ([0, 10]) = \llbracket [-15, -5], [0, 10] \rrbracket. \quad (\text{A.44})$$

As we can see, $\llbracket f \rrbracket ([x])$ has more information than $[f]([x], [p])$ because, as an example, $[y] = [-2, -1]$ is an output impossible to obtain.

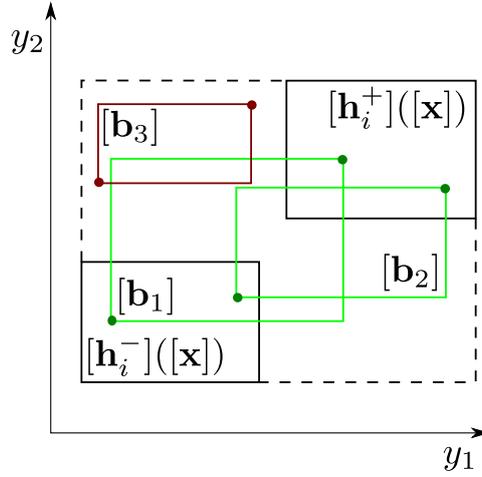


Figure A.8 – Illustration a thick box $\llbracket \mathbf{h}_i \rrbracket ([\mathbf{x}]) = \llbracket [\mathbf{h}_i^-]([\mathbf{x}]), [\mathbf{h}_i^+]([\mathbf{x}])] \rrbracket$ and boxes such that $[b_1] \in \llbracket \mathbf{h}_i \rrbracket ([\mathbf{x}])$ and $[b_2] \in \llbracket \mathbf{h}_i \rrbracket ([\mathbf{x}])$ but $[b_3] \notin \llbracket \mathbf{h}_i \rrbracket ([\mathbf{x}])$

Consider a thick box $\llbracket \mathbf{a} \rrbracket = \llbracket [\mathbf{a}^-], [\mathbf{a}^+] \rrbracket$ and a box $[\mathbf{b}] = [\underline{\mathbf{b}}, \overline{\mathbf{b}}]$. The following definitions are deduced from Proposition 7 introduced in (Desrochers and Jaulin, 2017) :

A thick box $\llbracket \mathbf{a} \rrbracket$ is included in a box $[\mathbf{b}]$, i.e., $\llbracket \mathbf{a} \rrbracket \subset [\mathbf{b}]$ if and only if **for all** coordinates $i \in \{1 \dots m\}$ the thick interval $\llbracket a_i \rrbracket$ is included in the interval $[b_i]$, where

$$\begin{aligned} \llbracket a_i \rrbracket \subset [b_i] &\Leftrightarrow \forall [a_i] \in \llbracket a_i \rrbracket = \llbracket [a_i^-], [a_i^+] \rrbracket, [a_i] \subset [b_i] \\ &\Leftrightarrow \begin{cases} \underline{b}_i - [a_i^-] \subset \mathbb{R}^- & \text{and} \\ [a_i^+] - \overline{b}_i \subset \mathbb{R}^- \end{cases} \end{aligned} \quad (\text{A.45})$$

A thick box $\llbracket \mathbf{a} \rrbracket$ is not included in a box $[\mathbf{b}]$, i.e., $\llbracket \mathbf{a} \rrbracket \not\subset [\mathbf{b}]$ if and only if **there exists** a coordinate $i \in \{1 \dots m\}$ such that the thick interval $\llbracket a_i \rrbracket$ is not included in the interval $[b_i]$, where

$$\begin{aligned} \llbracket a_i \rrbracket \not\subset [b_i] &\Leftrightarrow \forall [a_i] \in \llbracket a_i \rrbracket = \llbracket [a_i^-], [a_i^+] \rrbracket, [a_i] \not\subset [b_i] \\ &\Leftrightarrow \begin{cases} [a_i^-] - \underline{b}_i \subset \mathbb{R}^- & \text{or} \\ \overline{b}_i - [a_i^+] \subset \mathbb{R}^- \end{cases} \end{aligned} \quad (\text{A.46})$$

A thick box $\llbracket \mathbf{a} \rrbracket$ has a non-empty intersection with a box $[\mathbf{b}]$, i.e., $\llbracket \mathbf{a} \rrbracket \cap [\mathbf{b}] \neq \emptyset$ if and only if **for all** coordinates $i \in \{1 \dots m\}$ the thick interval $\llbracket a_i \rrbracket$ has a non-empty intersection with the interval $[b_i]$,

where

$$\begin{aligned} \llbracket a_i \rrbracket \cap [b_i] \neq \emptyset &\Leftrightarrow \forall [a_i] \in \llbracket a_i \rrbracket = \llbracket [a_i^-], [a_i^+] \rrbracket, [a_i] \cap [b_i] \neq \emptyset \\ &\Leftrightarrow \begin{cases} \underline{b}_i - [a_i^+] \in \mathbb{R}^- & \text{and} \\ [a_i^-] - \bar{b}_i \in \mathbb{R}^-. \end{cases} \end{aligned} \quad (\text{A.47})$$

A thick box $\llbracket \mathbf{a} \rrbracket$ has an empty intersection with a box $[\mathbf{b}]$, *i.e.*, $\llbracket \mathbf{a} \rrbracket \cap [\mathbf{b}] = \emptyset$ if and only if **there exists** a coordinate $i \in \{1 \dots m\}$ such that the thick interval $\llbracket a_i \rrbracket$ has an empty intersection with the interval $[b_i]$, where

$$\begin{aligned} \llbracket a_i \rrbracket \cap [b_i] = \emptyset &\Leftrightarrow \forall [a_i] \in \llbracket a_i \rrbracket = \llbracket [a_i^-], [a_i^+] \rrbracket, [a_i] \cap [b_i] = \emptyset \\ &\Leftrightarrow \begin{cases} [a_i^+] - \underline{b}_i \in \mathbb{R}^- & \text{or} \\ \bar{b}_i - [a_i^-] \in \mathbb{R}^-. \end{cases} \end{aligned} \quad (\text{A.48})$$

From those results, several properties can be obtained.

Proposition 28. For some $[x] \subset \mathbb{X}_0$, if $\llbracket \mathbf{h} \rrbracket ([x]) \cap [\mathbf{y}] = \emptyset$, then $[x] \cap \overline{\mathbb{X}}^{\exists} = \emptyset$

Démonstration. For some $[x] \subset \mathbb{X}_0$, if $\llbracket \mathbf{h} \rrbracket ([x]) \cap [\mathbf{y}] = \emptyset$, then, according to (A.20) and (A.34), one has $\llbracket \mathbf{h} \rrbracket ([x]) \cap [\mathbf{y}] = \emptyset$. Therefore, $\forall x \in [x], \forall f \in \llbracket \mathbf{h} \rrbracket, f(x) \notin [\mathbf{y}]$ which leads to have $[x] \cap \overline{\mathbb{X}}^{\exists} = \emptyset$ \square

Proposition 29. For some $[x] \subset \mathbb{X}_0$, if $\llbracket \mathbf{h} \rrbracket ([x]) \subset [\mathbf{y}]$, then $[x] \subset \overline{\mathbb{X}}^{\forall}$

Démonstration. For some $[x] \subset \mathbb{X}_0$, if $\llbracket \mathbf{h} \rrbracket ([x]) \subset [\mathbf{y}]$, then, according to (A.20) and (A.34), one has $\llbracket \mathbf{h} \rrbracket ([x]) \subset [\mathbf{y}]$. Therefore, $\forall x \in [x], \forall f \in \llbracket \mathbf{h} \rrbracket, f(x) \in [\mathbf{y}]$ which leads to have $[x] \subset \overline{\mathbb{X}}^{\forall}$ \square

Proposition 30. For some $[x] \subset \mathbb{X}_0$, if $\llbracket \mathbf{h} \rrbracket ([x]) \cap [\mathbf{y}] \neq \emptyset$, then $[x] \subset \overline{\mathbb{X}}^{\exists}$

Démonstration. Let assume that we have $\llbracket \mathbf{h} \rrbracket ([x]) \cap [\mathbf{y}] \neq \emptyset$ for a $[x] \subset \mathbb{X}_0$. Consider a $x \in [x]$, thus, $\llbracket \mathbf{h} \rrbracket (x)$ becomes a box. However, according to (A.47), we still have $\llbracket \mathbf{h} \rrbracket (x) \cap [\mathbf{y}] \neq \emptyset$. Consequently, according to (A.20) and (A.34), one has $\llbracket \mathbf{h} \rrbracket (x) \cap [\mathbf{y}] \neq \emptyset$. Therefore, using (A.27), there exists a $f \in \llbracket \mathbf{h} \rrbracket$ such that $f(x) \in [\mathbf{y}]$. Since this is true for any $x \in [x]$, one has $[x] \subset \overline{\mathbb{X}}^{\exists}$ \square

Proposition 31. For some $[x] \subset \mathbb{X}_0$, if $\llbracket \mathbf{h} \rrbracket ([x]) \not\subset [\mathbf{y}]$, then $[x] \not\subset \overline{\mathbb{X}}^{\forall}$

Démonstration. Let assume that we have $\llbracket \mathbf{h} \rrbracket ([x]) \not\subset [\mathbf{y}]$ for a $[x] \subset \mathbb{X}_0$. Consider a $x \in [x]$, thus, $\llbracket \mathbf{h} \rrbracket (x)$ becomes a box. However, according to (A.46), we still have $\llbracket \mathbf{h} \rrbracket (x) \not\subset [\mathbf{y}]$. Consequently, according to (A.20) and (A.34), one has $\llbracket \mathbf{h} \rrbracket (x) \not\subset [\mathbf{y}]$. Therefore, using (A.27), there exists a $f \in \llbracket \mathbf{h} \rrbracket$ such that $f(x) \notin [\mathbf{y}]$. Since, there exist a $x \in [x]$ and a $f \in \llbracket \mathbf{h} \rrbracket$ such that $f(x) \notin [\mathbf{y}]$, one cannot have $[x] \subset \overline{\mathbb{X}}^{\forall}$. Consequently, $[x] \not\subset \overline{\mathbb{X}}^{\forall}$. \square

A.4.3 . Thick SIVIA without $\underline{\mathbb{X}}^\vee$

Assuming a TIF $\llbracket \mathbf{h} \rrbracket$ of $\llbracket \mathbf{h} \rrbracket$ is available, one can obtain an outer-approximation of $\overline{\mathbb{X}}^\exists$ using the Thick SIVIA algorithm.

Similarly to SIVIA, the Propositions 30 and 28 are used to test if a box $[x] \subset \mathbb{X}_0$ belongs to $\overline{\mathbb{X}}^\exists$ or if it contains no solution. If one has $\llbracket \mathbf{h} \rrbracket ([x]) \cap [\mathbf{y}] \neq \emptyset$, then, according to Proposition 30, one has $[x] \subset \overline{\mathbb{X}}^\exists$. Else, if $\llbracket \mathbf{h} \rrbracket ([x]) \cap [\mathbf{y}] = \emptyset$, then, according to Proposition 28, one has $[x] \cap \overline{\mathbb{X}}^\exists = \emptyset$. Otherwise, the box is bisected and retested.

Consequently, the Thick SIVIA provides :

- a subpaving of boxes $\mathcal{B}_{\text{penumbra}}$ such that all $[x] \in \mathcal{B}_{\text{penumbra}}$ satisfy 30
- a subpaving of boxes \mathcal{B}_{out} such that all $[x] \in \mathcal{B}_{\text{out}}$ satisfy 28
- a subpaving of boxes $\mathcal{B}_{\text{border}}$ such that none $[x] \in \mathcal{B}_{\text{border}}$ satisfy 30 and 28 but are deemed too small to be bisected again

where $\mathcal{B}_{\text{penumbra}} \cup \mathcal{B}_{\text{border}}$ is an outer-approximation of $\overline{\mathbb{X}}^\exists$ which is a easier to obtain than an outer-approximation of $\overline{\mathbb{X}}^\exists$ with the SIVIA algorithm, but at the cost of some additional pessimism. Figure A.9 illustrates the characterization by Thick SIVIA of the set (A.21) when the vector of parameters \mathbf{p} is only known to belong to $[-0.5, 0.5]^2$.

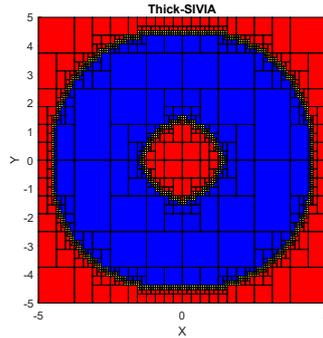


Figure A.9 – Characterization of the set defined by (A.21) using Thick SIVIA while considering an uncertain vector of parameters. $\mathcal{B}_{\text{penumbra}}$ is the 272 blue boxes. \mathcal{B}_{out} is the red boxes. $\mathcal{B}_{\text{border}}$ is the 912 yellow boxes.

A.4.4 . Thick SIVIA with $\underline{\mathbb{X}}^\vee$

Other tests can be defined to get an inner-approximation of $\underline{\mathbb{X}}^\vee$. According to Proposition 29, if we have $\llbracket \mathbf{h} \rrbracket ([x]) \subset [\mathbf{y}]$ for some $[x] \subset \mathbb{X}_0$, then we have $[x] \subset \underline{\mathbb{X}}^\vee$. Nevertheless, boxes satisfying Proposition 29 also satisfy Proposition 30. Therefore, to avoid an overlapping of boxes, one may prefer to approximate $\overline{\mathbb{X}}^\exists \setminus \underline{\mathbb{X}}^\vee$ rather than $\overline{\mathbb{X}}^\exists$. For that purpose, one must combine Proposition 30 with Proposition 31 to obtain the following test : if we have $\llbracket \mathbf{h} \rrbracket ([x]) \cap [\mathbf{y}] \neq \emptyset$ while having $\llbracket \mathbf{h} \rrbracket ([x]) \not\subset [\mathbf{y}]$ for a box $[x] \subset \mathbb{X}_0$, then we have respectively $[x] \subset \overline{\mathbb{X}}^\exists$ and $[x] \not\subset \underline{\mathbb{X}}^\vee$. Otherwise, if a box $[x]$ does not satisfy any test, then it is bisected and its children are retested using the same process.

Consequently, this Thick SIVIA provides :

- a subpaving of boxes \mathcal{B}_{in} such that all $[x] \in \mathcal{B}_{\text{in}}$ satisfy 29
- a subpaving of boxes \mathcal{B}_{out} such that all $[x] \in \mathcal{B}_{\text{out}}$ satisfy 28

- a subpaving of boxes $\mathcal{B}_{\text{penumbra}}$ such that all $[x] \in \mathcal{B}_{\text{penumbra}}$ satisfy 30 and 31
- a subpaving of boxes $\mathcal{B}_{\text{border}}$ such that none $[x] \in \mathcal{B}_{\text{border}}$ satisfy no test but are deemed too small to be bisected again

where \mathcal{B}_{in} is an inner-approximation of $\underline{\mathbb{X}}^{\forall}$, and $\mathcal{B}_{\text{in}} \cup \mathcal{B}_{\text{penumbra}} \cup \mathcal{B}_{\text{border}}$ is an outer-approximation of $\overline{\mathbb{X}}^{\exists}$. Figure A.10 illustrates the characterization by Thick SIVIA of the set (A.21) when the vector of parameters p is only known to belong to $[-0.5, 0.5]^2$.

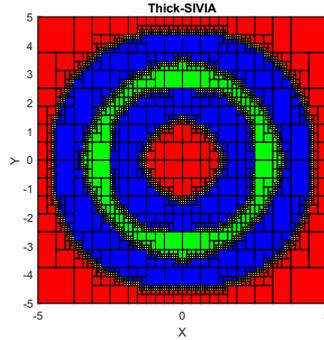


Figure A.10 – Characterization of the set defined by (A.21) using Thick SIVIA while considering an uncertain vector of parameters. \mathcal{B}_{in} is the 236 green boxes. $\mathcal{B}_{\text{penumbra}}$ is the 1752 blue boxes. \mathcal{B}_{out} is the red boxes. $\mathcal{B}_{\text{border}}$ is the 1752 yellow boxes.

Bibliographie

- Ajmera, Y. and Singh, S. P. (2020). Autonomous UAV-based Target Search, Tracking and Following using Reinforcement Learning and YOLOFlow. In *IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 15–20.
- Allik, B. (2019). Tracking of multiple targets across distributed platforms with fov constraints. In *Proc. IEEE CDC*, pages 6044–6049.
- Anderson, I. (2019). Heterogeneous map merging : State of the art. *Robotics*, 8(3) :74.
- Arras, K. O., Mozos, O. M., and Burgard, W. (2007). Using boosted features for the detection of people in 2d range data. In *Proceedings 2007 IEEE international conference on robotics and automation*, pages 3402–3407. IEEE.
- Asgharivaskasi, A. and Atanasov, N. (2023). Semantic OcTree mapping and Shannon mutual information computation for robot exploration. *IEEE Trans. Robotics*, 39(3) :1910–1928.
- Astrom, K. J. (1965). Optimal control of Markov processes with incomplete state information I. *Journal of mathematical analysis and applications*, 10 :174–205.
- Atanasov, N., Sankaran, B., Le Ny, J., Pappas, G. J., and Daniilidis, K. (2014). Nonmyopic view planning for active object classification and pose estimation. *IEEE Trans. Robotics*, 30(5) :1078–1090.
- Bai, X., Wang, Z., Zou, L., and Zhang, Z. (2021). Target tracking for wireless localization systems using set-membership filtering : A component-based event-triggered mechanism. *Automatica*, 132 :109795.
- Banerjee, A. and Schneider, J. (2024). Decentralized Multi-Agent Active Search and Tracking when Targets Outnumber Agents. *arXiv preprint arXiv :2401.03154*.
- Basar, T. and Olsder, G. J. (1998). *Dynamic noncooperative game theory*. SIAM.
- Batinovic, A., Ivanovic, A., Petrovic, T., and Bogdan, S. (2022). A shadowcasting-based next-best-view planner for autonomous 3D exploration. *IEEE Robotics and Automation Letters*, 7(2) :2969–2976.
- Bayer, J. and Faigl, J. (2019). On autonomous spatial exploration with small hexapod walking robot using tracking camera intel realsense t265. In *2019 European Conference on Mobile Robots (ECMR)*, pages 1–6. IEEE.
- Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-D shapes. In *Sensor fusion IV : control paradigms and data structures*, volume 1611, pages 586–606. Spie.
- Blanding, W., Koch, W., and Nickel, U. (2006). Tracking through jamming using negative information. In *2006 9th International Conference on Information Fusion*, pages 1–8. IEEE.
- Bochkovskiy, A. (2020). Yolov4 : Optimal speed and accuracy of object detection. *arXiv preprint arXiv :2004.10934*.
- Bouabdallah, S. (2007). Design and control of quadrotors with application to autonomous flying. Technical report, Epfl.

- Cai, Z.-S., Sun, L.-N., Gao, H.-B., Zhou, P.-C., Piao, S.-H., and Huang, Q.-C. (2008). Multi-robot cooperative pursuit based on task bundle auctions. In *Intelligent Robotics and Applications : First International Conference, ICIRA 2008, Wuhan, China, October 15-17, 2008, Proceedings, Part I 1*, pages 235–244. Springer.
- Cannon, M., Cheng, Q., Kouvaritakis, B., and Rakovic, S. V. (2012). Stochastic tube MPC with state estimation. *Automatica*, 48(3) :536–541.
- Cao, Y., Wu, Z., and Shen, C. (2017). Estimating depth from monocular images as classification using deep fully convolutional residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(11) :3174–3182.
- Cheng, K., Hu, T., Wu, D., Li, T., Wang, S., Liu, K., Tian, Z., and Yi, D. (2024). Heterogeneous UAV Swarm Collaborative Search Mission Path Optimization Scheme for Dynamic Targets. *International Journal of Aerospace Engineering*, 2024(1) :6643424.
- Christofides, P. D., Scattolini, R., De La Pena, D. M., and Liu, J. (2013). Distributed model predictive control : A tutorial review and future research directions. *Comput Chem Eng*, 51 :21–41.
- Cichella, V., Kaminer, I., Dobrokhodov, V., and Hovakimyan, N. (2015). Coordinated vision-based tracking for multiple UAVs. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 656–661. IEEE.
- Dames, P. M. (2020). Distributed multi-target search and tracking using the PHD filter. *Autonomous robots*, 44(3) :673–689.
- Dang, X., Shao, C., and Hao, Z. (2019). Target detection coverage algorithm based on 3D-Voronoi partition for three-dimensional wireless sensor networks. *Mobile Information Systems*, 2019.
- Dellenbach, P., Deschaud, J.-E., Jacquet, B., and Goulette, F. (2022). CT-ICP : Real-time elastic LiDAR odometry with loop closure. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5580–5586. IEEE.
- Desrochers, B. and Jaulin, L. (2017). Thick set inversion. *Artificial Intelligence*, 249 :1–18.
- Di Gennaro, T. M. and Waldmann, J. (2023). Sensor Fusion with Asynchronous Decentralized Processing for 3D Target Tracking with a Wireless Camera Network. *Sensors*, 23(3) :1194–1228.
- DJI. <https://enterprise.dji.com/drone-rescue-map/>.
- Drevelle, V. and Bonnifait, P. (2011). A set-membership approach for high integrity height-aided satellite positioning. *GPS solutions*, 15(4) :357–368.
- Du, W., Gu, F., Zhou, H., Di, C., Xiong, J., and He, Y. (2018). Air-Ground Cooperative Environment Modeling with Bounded Elevation Map. In *in Proc. IEEE ICIST*, pages 422–428.
- Ehambram, A., Jaulin, L., and Wagner, B. (2022). Hybrid Interval-Probabilistic Localization in Building Maps. *IEEE Robotics and Automation Letters*, 7(3) :7059–7066.
- Fan, Z., Zhu, Y., He, Y., Sun, Q., Liu, H., and He, J. (2022). Deep learning on monocular object pose detection and tracking : A comprehensive overview. *ACM Computing Surveys*, 55(4) :1–40.
- Fankhauser, P., Bloesch, M., Gehring, C., Hutter, M., and Siegwart, R. (2014). Robot-centric elevation mapping with uncertainty estimates. In *Mobile Service Robotics*, pages 433–440. World Scientific.

- Faugeras, O. (1993). *Three-dimensional computer vision : a geometric viewpoint*. MIT press.
- Fei, B., Bao, W., Zhu, X., Liu, D., Men, T., and Xiao, Z. (2022). Autonomous cooperative search model for multi-UAV with limited communication network. *IEEE Internet of Things Journal*, 9(19) :19346–19361.
- Frew, E. W. and Elston, J. (2008). Target assignment for integrated search and tracking by active robot networks. In *2008 IEEE International Conference on Robotics and Automation*, pages 2354–2359. IEEE.
- Furukawa, T., Bourgault, F., Lavis, B., and Durrant-Whyte, H. F. (2006). Recursive Bayesian search-and-tracking using coordinated UAVs for lost targets. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2521–2526. IEEE.
- Geller, D. K. and Oshman, Y. (2014). Negative Collision Measurements in Position/Velocity State Estimation. *Journal of Guidance, Control, and Dynamics*, 37(3) :1015–1022.
- Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.
- Goldhoorn, A., Garrell, A., Alquézar, R., and Sanfeliu, A. (2018). Searching and tracking people with cooperative mobile robots. *Autonomous Robots*, 42(4) :739–759.
- Grilli, E., Menna, F., and Remondino, F. (2017). A review of point clouds segmentation and classification algorithms. *ISPRS Archives*, 42 :339.
- Hardouin, G., Moras, J., Morbidi, F., Marzat, J., and Mouaddib, E. M. (2020). Next-Best-View planning for surface reconstruction of large-scale 3D environments with multiple UAVs. In *Proc. IEEE IROS*, pages 1567–1574.
- Haugen, J. and Imsland, L. (2015). Monitoring moving objects using aerial mobile sensors. *IEEE Transactions on Control Systems Technology*, 24(2) :475–486.
- Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap : An efficient probabilistic 3D mapping framework based on octrees. *Autonomous robots*, 34 :189–206.
- Hou, J., Li, J., Ni, Y., and Dong, W. (2021). Unmanned Aerial Vehicle's State Estimation with Multiple Unmanned Ground Vehicles Cooperative Observation Based on Set-Membership Filter. In *2021 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pages 411–416. IEEE.
- Hou, Y., Zhao, J., Zhang, R., Cheng, X., and Yang, L. (2023). UAV Swarm Cooperative Target Search : A Multi-Agent Reinforcement Learning Approach. *IEEE Trans. Intelligent Vehicles*, 9(1) :568–578.
- How, J. P., Fraser, C., Kulling, K. C., Bertuccelli, L. F., Toupet, O., Brunet, L., Bachrach, A., and Roy, N. (2009). Increasing autonomy of UAVs. *IEEE Robotics & Automation Magazine*, 16(2) :43–51.
- Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., and others (2019). Searching for mobilenetv3. In *Proc. IEEE/CVF ICCV*, pages 1314–1324.
- Hu, C., Zhang, Z., Yang, N., Shin, H.-S., and Tsourdos, A. (2019). Fuzzy multiobjective cooperative surveillance of multiple UAVs based on distributed predictive control for unknown ground moving target in urban environment. *Aerospace science and technology*, 84 :329–338.
- Ibenthal, J. (2022). *Multi-target tracking by non-linear set-membership methods*. PhD Thesis, Université Paris-Saclay.

- Ibenthal, J., Kieffer, M., Meyer, L., Piet-Lahanier, H., and Reynaud, S. (2021). Bounded-error target localization and tracking using a fleet of UAVs. *Automatica*, 132 :109809–109824.
- Ibenthal, J., Meyer, L., Piet-Lahanier, H., and Kieffer, M. (2020). Target search and tracking using a fleet of UAVs in presence of decoys and obstacles. In *IEEE CDC*, pages 188–194. IEEE.
- Ibenthal, J., Meyer, L., Piet-Lahanier, H., and Kieffer, M. (2023). Localization of Partially Hidden Moving Targets Using a Fleet of UAVs via Bounded-Error Estimation. *IEEE Trans. Robotics*, 39(6) :4211–4229.
- Jamwal, N., Jindal, N., and Singh, K. (2016). A survey on depth map estimation strategies. In *ICSP*, pages 1–5. IET.
- Jaulin, L. (2009). Robust set-membership state estimation; application to underwater robotics. *Automatica*.
- Jaulin, L., Kieffer, M., Didrit, O., and Walter, E. (2001). *Applied Interval analysis*. Springer.
- Jaulin, L. and Walter, E. (1993). Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4) :1053–1064. Number : 4.
- Jeon, H.-G., Park, J., Choe, G., Park, J., Bok, Y., Tai, Y.-W., and So Kweon, I. (2015). Accurate depth map estimation from a lenslet light field camera. In *Proc. IEEE CVPR*, pages 1547–1555.
- Ji, Y., Zhao, Y., Chen, B., Zhu, Z., Liu, Y., Zhu, H., and Qiu, S. (2022). Source searching in unknown obstructed environments through source estimation, target determination, and path planning. *Building and Environment*, 221 :109266–109306.
- Jiang, P., Deng, X., Wan, S., Qi, H., and Zhang, S. (2023). Confidence-enhanced mutual knowledge for uncertain segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 25(1) :725–737.
- Jiang, P., Ergu, D., Liu, F., Cai, Y., and Ma, B. (2022). A Review of Yolo algorithm developments. *Procedia computer science*, 199 :1066–1073.
- Jin, L., Cheng, J., and Huang, H. (2010). Human tracking in the complicated background by particle filter using color-histogram and hog. In *2010 International Symposium on Intelligent Signal Processing and Communication Systems*, pages 1–4. IEEE.
- Johnson, J. (1985). Analysis of image forming systems. *Selected papers on infrared design. Part I and II*, 513 :761.
- Kenmogne, I.-F., Drevelle, V., and Marchand, E. (2019). Cooperative localization of drones by using interval methods. *Acta Cybernetica*, pages 1–16.
- Kenmogne Fokam, I. F. (2019). *Localisation ensembliste de drones à l'aide de méthodes par intervalles*. PhD Thesis, Rennes 1.
- Khan, A., Rinner, B., and Cavallaro, A. (2016). Cooperative robots to observe moving targets. *IEEE transactions on cybernetics*, 48(1) :187–198.
- Kieffer, M. (1999). *Estimation ensembliste par analyse par intervalles Application à la localisation d'un véhicule*. PhD Thesis, Université Paris Sud-Paris XI.
- Kim, J. and Crassidis, J. L. (2010). UAV path planning for maximum visibility of ground targets in an urban area. In *2010 13th International Conference on Information Fusion*, pages 1–7. IEEE.

- Koopman, B. O. (1956). The theory of search. I. Kinematic bases. *Operations research*, 4(3) :324–346.
- Kuhlman, M. J., Otte, M. W., Sofge, D., and Gupta, S. K. (2017). Multipass target search in natural environments. *Sensors*, 17(11) :2514–2550.
- Lauri, M., Hsu, D., and Pajarinen, J. (2022). Partially observable markov decision processes in robotics : A survey. *IEEE Transactions on Robotics*, 39(1) :21–40.
- Le, V. T. H., Stoica, C., Dumur, D., Alamo, T., and Camacho, E. F. (2011). Robust tube-based constrained predictive control via zonotopic set-membership estimation. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 4580–4585. IEEE.
- Li, J., Zhai, X., Xu, J., and Li, C. (2021). Target search algorithm for AUV based on real-time perception maps in unknown environment. *Machines*, 9(8) :147–173.
- Li, J., Zhang, J., Zhang, G., and Zhang, B. (2018). An adaptive prediction target search algorithm for multi-AUVs in an unknown 3D environment. *Sensors*.
- Li, Y., Zhang, Z., Sun, Q., and Huang, Y. (2024). A distributed framework for multiple UAV cooperative target search under dynamic environment. *Journal of the Franklin Institute*, page 106810.
- Liu, D., Bao, W., Zhu, X., Fei, B., Xiao, Z., and Men, T. (2022a). Vision-aware air-ground cooperative target localization for UAV and UGV. *Aerosp. Sci. Technol.*, 124 :107525–107540.
- Liu, D., Zhu, X., Bao, W., Fei, B., and Wu, J. (2022b). SMART : Vision-Based Method of Cooperative Surveillance and Tracking by Multiple UAVs in the Urban Environment. *IEEE Transactions on Intelligent Transportation Systems*, 23(12) :24941–24956.
- Liu, J., Chen, X., Xiao, J., Lin, S., Zheng, Z., and Lu, H. (2023). Hybrid Map-Based Path Planning for Robot Navigation in Unstructured Environments. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2216–2223. IEEE.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. (2016). Ssd : Single shot multibox detector. In *Computer Vision-ECCV 2016 : 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I 14*, pages 21–37. Springer.
- Liu, Y., Liu, H., Tian, Y., and Sun, C. (2020). Reinforcement learning based two-level control framework of UAV swarm for cooperative persistent surveillance in an unknown urban area. *Aerospace Science and Technology*, 98 :105671.
- Liu, Y., Wang, Q., Hu, H., and He, Y. (2018). A novel real-time moving target tracking and path planning system for a quadrotor UAV in unknown unstructured outdoor scenes. *IEEE Trans. Syst. Man. Cybern.*, 49(11) :2362–2372.
- Liu, Y., Xu, M., Jiang, G., Tong, X., Yun, J., Liu, Y., Chen, B., Cao, Y., Sun, N., and Li, Z. (2022c). Target localization in local dense mapping using RGBD SLAM and object detection. *Concurrency and Computation : Practice and Experience*, 34(4) :e6655.
- Lopez, B. T., Slotine, J.-J. E., and How, J. P. (2019). Dynamic tube MPC for nonlinear systems. In *2019 American Control Conference (ACC)*, pages 1655–1662. IEEE.
- Lu, K. and Zhou, R. (2017). Multi-sensor fusion for robust target tracking in the simultaneous presence of set-membership and stochastic Gaussian uncertainties. *IET Radar, Sonar & Navigation*, 11(4) :621–628.

- Luo, Q., Luan, T. H., Shi, W., and Fan, P. (2023). Edge Computing Enabled Energy-Efficient Multi-UAV Cooperative Target Search. *IEEE Transactions on Vehicular Technology*, 72(6) :7757–7771.
- Luo, X., Wu, Y., and Zhao, L. (2022). YOLOD : A target detection method for UAV aerial imagery. *Remote Sensing*, 14(14) :3240.
- Lyu, M., Zhao, Y., Huang, C., and Huang, H. (2023). Unmanned aerial vehicles for search and rescue : A survey. *Remote Sensing*, 15(13) :3266.
- Madhuanand, L., Nex, F., and Yang, M. Y. (2021). Self-supervised monocular depth estimation from oblique UAV videos. *ISPRS journal of photogrammetry and remote sensing*, 176 :1–14.
- Meera, A. A., Popovic, M., Millane, A., and Siegwart, R. (2019). Obstacle-aware adaptive informative path planning for uav-based target search. In *Proc. IEEE ICRA*, pages 718–724.
- Mendes, E., Koch, P., and Lacroix, S. (2016). ICP-based pose-graph SLAM. In *2016 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 195–200. IEEE.
- Merhy, D. (2019). *Contribution to ellipsoidal and zonotopic set-membership state estimation*. PhD Thesis, Université Paris Saclay (COMUE).
- Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., and Terzopoulos, D. (2021). Image segmentation using deep learning : A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(7) :3523–3542.
- Minaeian, S., Liu, J., and Son, Y.-J. (2015). Vision-based target detection and localization via a team of cooperative UAV and UGVs. *IEEE Trans. Syst. Man. Cybern.*, 46(7) :1005–1016.
- Minaeian, S., Liu, J., and Son, Y.-J. (2018). Effective and efficient detection of moving targets from a UAV camera. *IEEE Trans. Intell. Transp. Syst.*, 19(2) :497–506.
- Moore, R. E. (1979). *Methods and applications of interval analysis*. SIAM.
- Mousavian, A., Anguelov, D., Flynn, J., and Kosecka, J. (2017). 3d bounding box estimation using deep learning and geometry. In *Pro. IEEE CVPR*, pages 7074–7082.
- Niedzielski, T., Jurecka, M., Mizinski, B., Pawul, W., and Motyl, T. (2021). First successful rescue of a lost person using the human detection system : A case study from Beskid Niski (SE Poland). *Remote Sensing*, 13(23) :4903–4921.
- Oleynikova, H., Taylor, Z., Fehr, M., Siegwart, R., and Nieto, J. (2017). Voxblox : Incremental 3d euclidean signed distance fields for on-board mav planning. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1366–1373. IEEE.
- Olsen, T., Stiffler, N. M., and O’Kane, J. M. (2022). Robust-by-design plans for multi-robot pursuit-evasion. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10716–10722. IEEE.
- Oron, S., Dekel, T., Xue, T., Freeman, W. T., and Avidan, S. (2017). Best-buddies similarity - Robust template matching using mutual nearest neighbors. *IEEE transactions on pattern analysis and machine intelligence*, 40(8) :1799–1813.
- O’rourke, J. and others (1987). *Art gallery theorems and algorithms*, volume 57. Oxford University Press Oxford.

- Pan, W.-T. (2012). A new fruit fly optimization algorithm : taking the financial distress model as an example. *Knowledge-Based Systems*, 26 :69–74.
- Park, J. and Kim, Y. (2012). Stereo vision based collision avoidance of quadrotor UAV. In *Proc. ICCAS*, pages 173–178.
- Parker, L. E. and Emmons, B. A. (1997). Cooperative multi-robot observation of multiple moving targets. In *Proceedings of International Conference on Robotics and Automation*, volume 3, pages 2082–2089. IEEE.
- Placed, J. A., Strader, J., Carrillo, H., Atanasov, N., Indelman, V., Carlone, L., and Castellanos, J. A. (2023). A survey on active simultaneous localization and mapping : State of the art and new frontiers. *IEEE Trans. Robotics*, 39(3) :1686–1705.
- Popović, M., Vidal-Calleja, T., Hitz, G., Sa, I., Siegwart, R., and Nieto, J. (2017). Multiresolution mapping and informative path planning for uav-based terrain monitoring. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1382–1388. IEEE.
- Queralta, J. P., Taipalmaa, J., Pullinen, B. C., Sarker, V. K., Gia, T. N., Tenhunen, H., Gabbouj, M., Raitoharju, J., and Westerlund, T. (2020). Collaborative multi-robot search and rescue : Planning, coordination, perception, and active vision. *IEEE Access*, 8 :191617–191643.
- Ranftl, R., Vineet, V., Chen, Q., and Koltun, V. (2016). Dense monocular depth estimation in complex dynamic scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4058–4066.
- Reboul, L., Kieffer, M., Piet-Lahanier, H., and Reynaud, S. (2019). Cooperative guidance of a fleet of UAVs for multi-target discovery and tracking in presence of obstacles using a set membership approach. *IFAC-PapersOnLine*, 52(12) :340–345.
- Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once : Unified, real-time object detection. In *Proc. IEEE CVPR*, pages 779–788.
- Ren, S., He, K., Girshick, R., and Sun, J. (2016). Faster R-CNN : Towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6) :1137–1149.
- Ribaric, S., Adrinek, G., and Segvic, S. (2004). Real-time active visual tracking system. In *Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference (IEEE Cat. No. 04CH37521)*, volume 1, pages 231–234. IEEE.
- Robin, C. and Lacroix, S. (2016). Multi-robot target detection and tracking : taxonomy and survey. *Autonomous Robots*, 40(4) :729–760.
- Sahin, C., Garcia-Hernando, G., Sock, J., and Kim, T.-K. (2020). A review on object pose recovery : from 3d bounding box detectors to full 6d pose estimators. *Image. Vision. Comput.*, 96 :103898.
- Schwenzer, M., Ay, M., Bergs, T., and Abel, D. (2021). Review on model predictive control : An engineering perspective. *The International Journal of Advanced Manufacturing Technology*, 117(5) :1327–1349.
- Schweppe, F. (1968). Recursive state estimation : Unknown but bounded errors and system inputs. *IEEE Transactions on Automatic Control*, 13(1) :22–28.

- Serdel, Q., Marzat, J., and Moras, J. (2023). SMaNa : Semantic Mapping and Navigation Architecture for Autonomous Robots. *Proc. ICINCO*, 1 :453–464.
- Shi, L., Marcano, N. J. H., and Jacobsen, R. H. (2021). A review on communication protocols for autonomous unmanned aerial vehicles for inspection application. *Microprocessors and Microsystems*, 86 :104340.
- Shimada, T., Nishikawa, H., Kong, X., and Tomiyama, H. (2023). Fast and High-Quality Monocular Depth Estimation with Optical Flow for Autonomous Drones. *Drones*, 7(2) :134–151.
- Shinde, S., Kothari, A., and Gupta, V. (2018). YOLO based human action recognition and localization. *Procedia computer science*, 133 :831–838.
- Silva, S. H., Rad, P., Beebe, N., Choo, K.-K. R., and Umapathy, M. (2019). Cooperative unmanned aerial vehicles with privacy preserving deep vision for real-time object identification and tracking. *Journal of parallel and distributed computing*.
- Souza, A. and Goncalves, L. M. (2016). Occupancy-elevation grid : an alternative approach for robotic mapping and navigation. *Robotica*, 34(11) :2592–2609.
- Souza, A. and Goncalves, L. M. (2018). Path planning based on traversability evaluation from occupancy-elevation grid maps. In *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, pages 308–313. IEEE.
- Stelzer, A., Hirschmüller, H., and Görner, M. (2012). Stereo-vision-based navigation of a six-legged walking robot in unknown rough terrain. *The International Journal of Robotics Research*, 31(4) :381–402.
- Sun, J., Li, B., Jiang, Y., and Wen, C.-y. (2016). A camera-based target detection and positioning UAV system for search and rescue (SAR) purposes. *Sensors*, 16(11) :1778–1802.
- Sun, W. and Tsiotras, P. (2017). Sequential pursuit of multiple targets under external disturbances via Zermelo–Voronoi diagrams. *Automatica*, 81 :253–260.
- Symington, A., Waharte, S., Julier, S., and Trigoni, N. (2010). Probabilistic target detection by camera-equipped UAVs. In *Proc. IEEE ICRA*, pages 4076–4081.
- Tang, H., Sun, W., Yu, H., Lin, A., Xue, M., and Song, Y. (2019). A novel hybrid algorithm based on PSO and FOA for target searching in unknown environments. *Applied Intelligence*, 49 :2603–2622.
- Tang, W., Wang, Z., Zhang, Q., and Shen, Y. (2020). Set-membership estimation for linear time-varying descriptor systems. *Automatica*, 115 :108867.
- Tang, X., Chen, X., McDonald, M., Mahler, R., Tharmarasa, R., and Kirubarajan, T. (2015). A multiple-detection probability hypothesis density filter. *IEEE Transactions on Signal Processing*, 63(8) :2007–2019.
- Tomono, M. (2009). Robust 3D SLAM with a stereo camera based on an edge-point ICP algorithm. In *2009 IEEE international conference on robotics and automation*, pages 4306–4311. IEEE.
- Van Tilburg, C. (2017). First report of using portable unmanned aircraft systems (drones) for search and rescue. *Wilderness & environmental medicine*, 28(2) :116–118.
- Vanegas, F., Campbell, D., Eich, M., and Gonzalez, F. (2016). UAV based target finding and tracking in GPS-denied and cluttered environments. In *Proc. IEEE IROS*, pages 2307–2313.

- Vanegas, F., Gaston, K. J., Roberts, J., and Gonzalez, F. (2019). A framework for UAV navigation and exploration in GPS-denied environments. In *IEEE Aerospace Conference*, pages 1–6.
- Vetrella, A. R., Fasano, G., and Accardo, D. (2016). Cooperative navigation in GPS-challenging environments exploiting position broadcast and vision-based tracking. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 447–456. IEEE.
- Vidal, R., Shakernia, O., Kim, H. J., Shim, D. H., and Sastry, S. (2002). Probabilistic pursuit-evasion games : theory, implementation, and experimental evaluation. *IEEE transactions on robotics and automation*, 18(5) :662–669.
- Vo, B.-N. and Ma, W.-K. (2006). The Gaussian mixture probability hypothesis density filter. *IEEE Transactions on signal processing*, 54(11) :4091–4104.
- Vo, B.-T., Vo, B.-N., and Cantoni, A. (2008). Bayesian filtering with random finite set observations. *IEEE Transactions on signal processing*, 56(4) :1313–1326.
- Wang, X., Liu, J., and Zhou, Q. (2016). Real-time multi-target localization from unmanned aerial vehicles. *Sensors*, 17(1) :33–61.
- Wang, X., Wang, S., and Ma, J.-J. (2007). An improved particle filter for target tracking in sensor systems. *Sensors*, 7(1) :144–156.
- Webots. <http://www.cyberbotics.com>.
- Wei, D., Zhang, L., Liu, Q., Chen, H., and Huang, J. (2024). UAV Swarm Cooperative Dynamic Target Search : A MAPPO-Based Discrete Optimal Control Method. *Drones*, 8(6) :214.
- Williams, J., Jiang, S., O'Brien, M., Wagner, G., Hernandez, E., Cox, M., Pitt, A., Arkin, R., and Hudson, N. (2020). Online 3D frontier-based UGV and UAV exploration using direct point cloud visibility. In *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 263–270. IEEE.
- Wu, Y., Wu, S., and Hu, X. (2020). Cooperative path planning of UAVs & UGVs for a persistent surveillance task in urban environments. *IEEE Internet of Things Journal*, 8(6) :4906–4919.
- Xie, W., Wang, L., Bai, B., Peng, B., and Feng, Z. (2019). An improved algorithm based on particle filter for 3D UAV target tracking. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE.
- Yadav, P. K. and Meng, W. (2014). Mobile targets localization in a field area using moving Gaussian peaks and probability map. In *11th IEEE International Conference on Control & Automation (ICCA)*, pages 416–421. IEEE.
- Yanmaz, E. (2023). Joint or decoupled optimization : Multi-UAV path planning for search and rescue. *Ad Hoc Networks*, 138 :103018–103031.
- Zagar, M., Kieffer, M., Piet-Lahanier, H., and Meyer, L. (2022). Robust collaborative 3D target localization using UAVs with state uncertainties. *IFAC-PapersOnLine*.
- Zeng, Z., Zhou, Y., Jenkins, O. C., and Desingh, K. (2018). Semantic mapping with simultaneous object detection and localization. In *Proc. IEEE IROS*, pages 911–918.

- Zhang, B., Lin, X., Zhu, Y., Tian, J., and Zhu, Z. (2024). Enhancing Multi-UAV Reconnaissance and Search Through Double Critic DDPG With Belief Probability Maps. *IEEE Trans. Intelligent Vehicles*, 9(2) :3827–3842.
- Zhang, H., Xie, C., Toriya, H., Shishido, H., and Kitahara, I. (2023). Vehicle Localization in a Completed City-Scale 3D Scene Using Aerial Images and an On-Board Stereo Camera. *Remote Sensing*, 15(15) :3871–3891.
- Zhang, S., Zhang, X., Li, T., Yuan, J., and Fang, Y. (2022). Fast active aerial exploration for traversable path finding of ground robots in unknown environments. *IEEE Trans. Instrum. Meas.*, 71 :1–13.
- Zhao, L., Li, R., Han, J., and Zhang, J. (2022). A distributed model predictive control-based method for multidifferent-target search in unknown environments. *IEEE Trans. Evol. Comput.*, 27(1) :111–125.
- Zhen, Z., Chen, Y., Wen, L., and Han, B. (2020). An intelligent cooperative mission planning scheme of UAV swarm in uncertain dynamic environment. *Aerospace Science and Technology*, 100 :105826–105842.
- Zhong, F., Wang, S., Zhang, Z., and Wang, Y. (2018). Detect-SLAM : Making object detection and SLAM mutually beneficial. In *IEEE WACV*, pages 1001–1010.
- Zhou, L., Leng, S., Liu, Q., and Wang, Q. (2021). Intelligent UAV swarm cooperation for multiple targets tracking. *IEEE Internet of Things Journal*, 9(1) :743–754.
- Zhu, X., Vanegas, F., Gonzalez, F., and Sanderson, C. (2021). A multi-UAV system for exploration and target finding in cluttered and GPS-denied environments. In *Proc. IEEE ICUAS*, pages 721–729.