

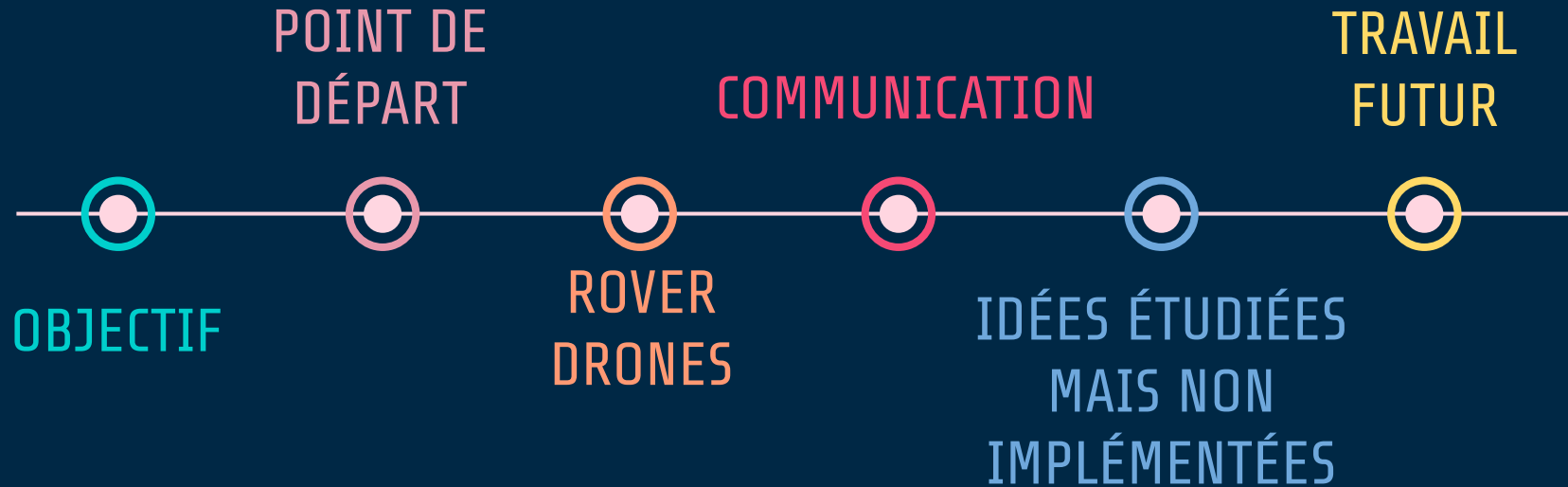


Challenge Intelligence répartie Robots en essaim

Groupe 7

Pau BECERRA
Matheus MONTEVERDE
Mateus RICCI
Fabrício VELLONE

Plan



OBJECTIF

Conduire une mission de cartographie et de détection de danger et de personnes dans un territoire hostile, inconnue, afin de préparer l'arrivée d'une équipe d'intervention (militaire ou secours)

POINT DE DÉPART

- Chaque équipe disposera d'une flotte de 10 drones et 4 robots terrestres.
- La mission : La découverte consistera en la détection puis l'identification d'une personne.
 - La détection est faite par drones.
 - L'identification est faite par un robot terrestre (arriver à moins d'une distance seuil de la personne).
- Il est possible que certains drones ou robots soient mis hors d'état de fonctionner pendant la mission. La flotte de drones et de robots terrestres doit être capable de se réorganiser en temps réel, afin de poursuivre le mission.

ROVER - Husky

Pour bouger le rover on a utilisé un paquet de ROS appelé "move_base".

Ce paquet permet de créer les cartes de coûts qui, en utilisant les capteurs du rover, permettent de définir la présence des obstacles par rapport à la position du robot.

Étant donné qu'il n'y a pas de cartographie, nous ne pouvons pas utiliser les capacités globales mais nous pouvons utiliser les capacités locales. Dans notre cas, nous avons utilisé la carte de coûts locale (local costmap) et la planification de trajectoire locale (local path planning).

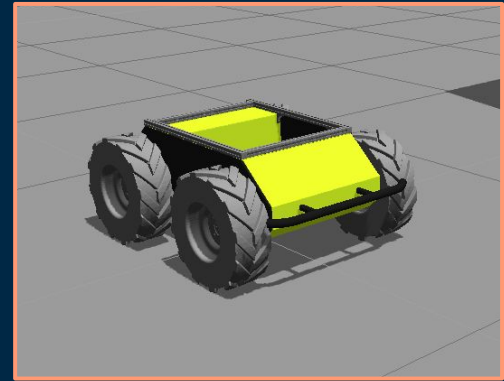


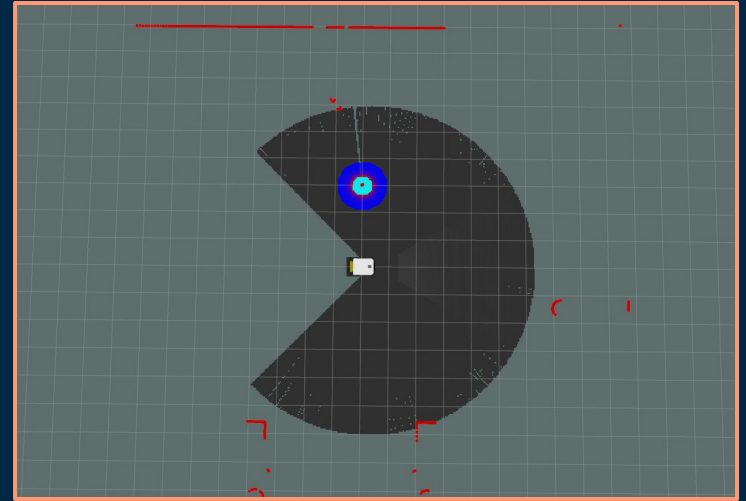
Image générique du modèle de rover "husky"

ROVER - Local costmap

La carte de coûts utilise l'information prise par les capteurs pour créer une "occupancy grid".

Cette "occupancy grid" définit:

- Les endroits dans l'espace où il existe des obstacles (violet)
- Il existe un risque de collision selon l'orientation du robot (bleu foncé)
- Les endroits libres d'obstacles (noir).
- Le bleu clair est pour les endroits inconnus.

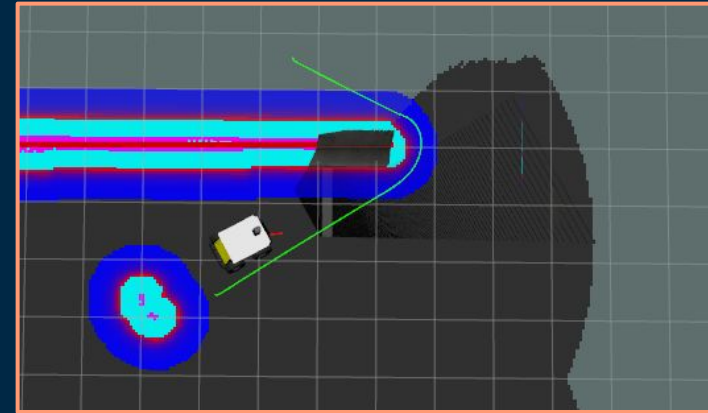


Local costmap dans notre simulation
(avec les mesures du laser en rouge)

ROVER - Local path planning

Une fois que la carte des coûts a été définie, l'algorithme de "path planning" utilise cette information pour calculer la trajectoire du robot en évitant les obstacles (la ligne verte dans l'image).

La ligne rouge devant le rover indique la direction que le robot doit prendre pour s'adapter à la trajectoire définie.

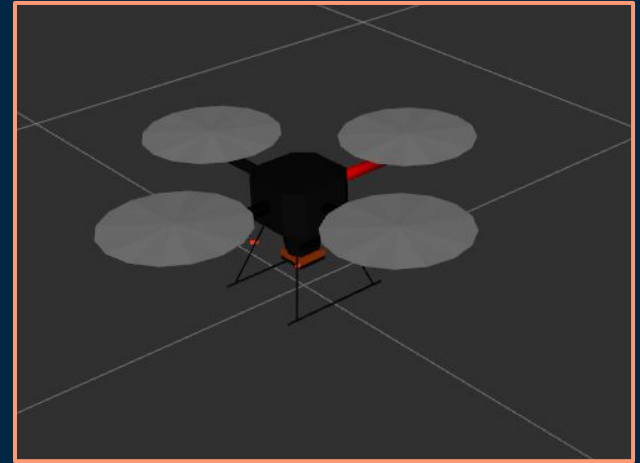


Le chemin calculé (vert) pour éviter les obstacles détectés par le laser

DRONES – Hector quadrotor

Comme 10 drones ont été prévus pour la mission de détection et d'identification de personnes hostiles, ceux-ci ont été répartis en 3 groupes :

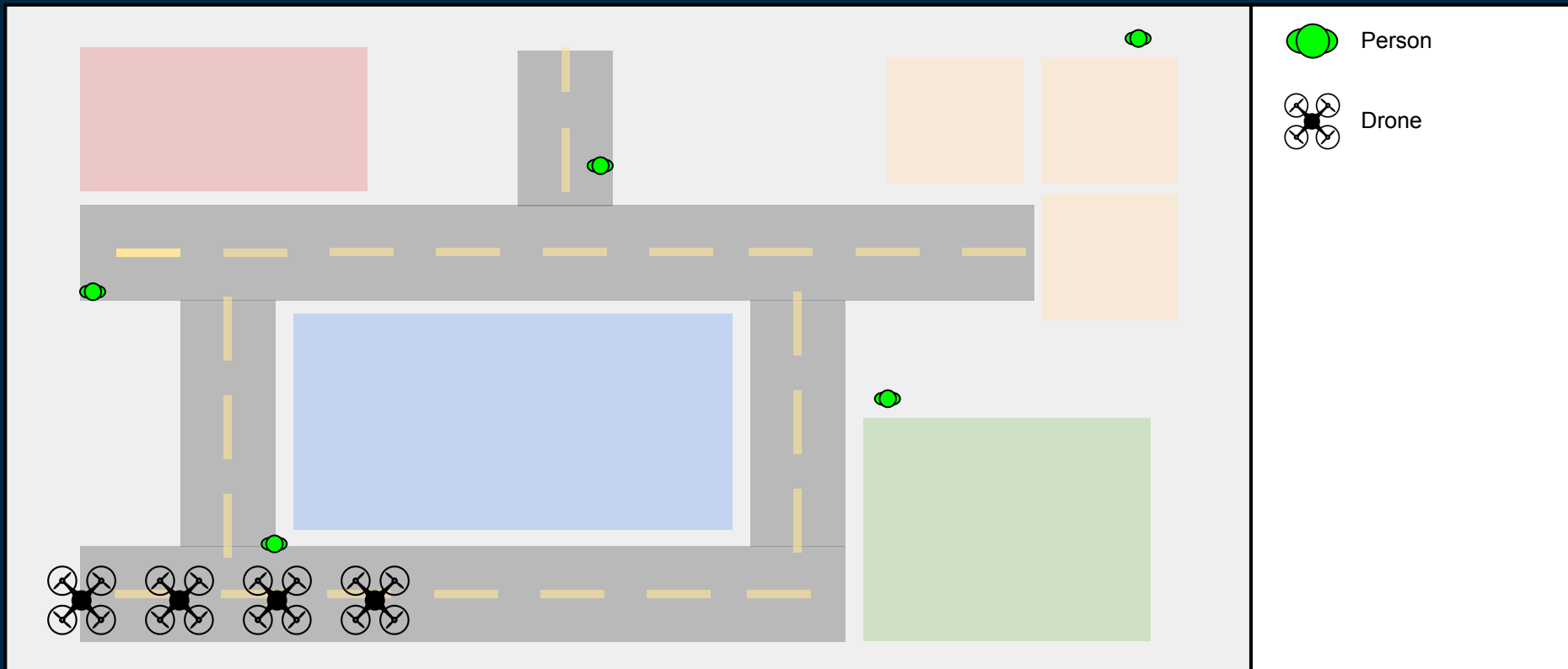
1. Reconnaissance de l'environnement et détection approximative des personnes (4 drones)
2. Précision de la position des personnes trouvées (4 drones)
3. Backup pour la reconnaissance, en cas de drones détruits (2 drones)



Le modèle "hector_quadrotor"



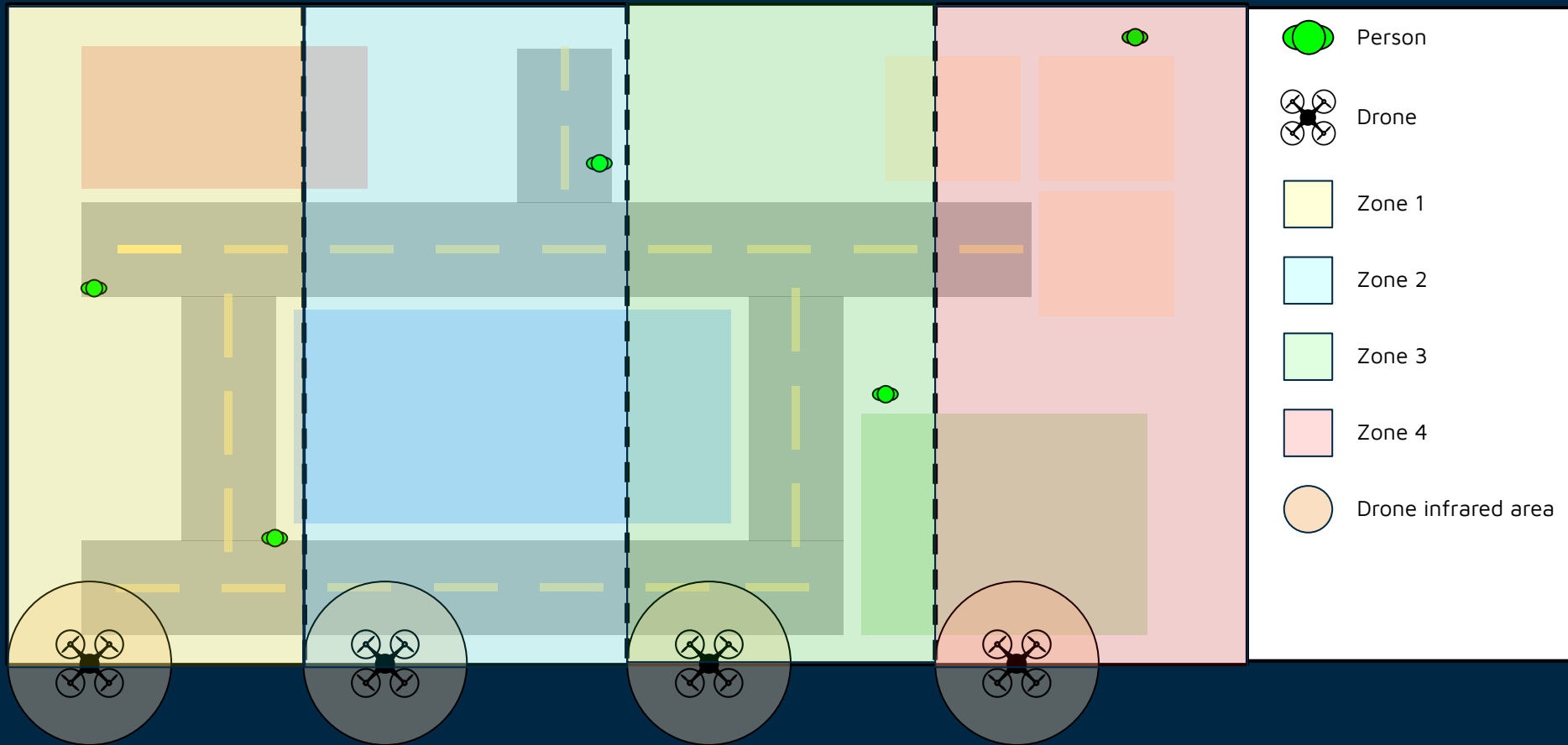
Stratégie pour le reconnaissance de l'environnement



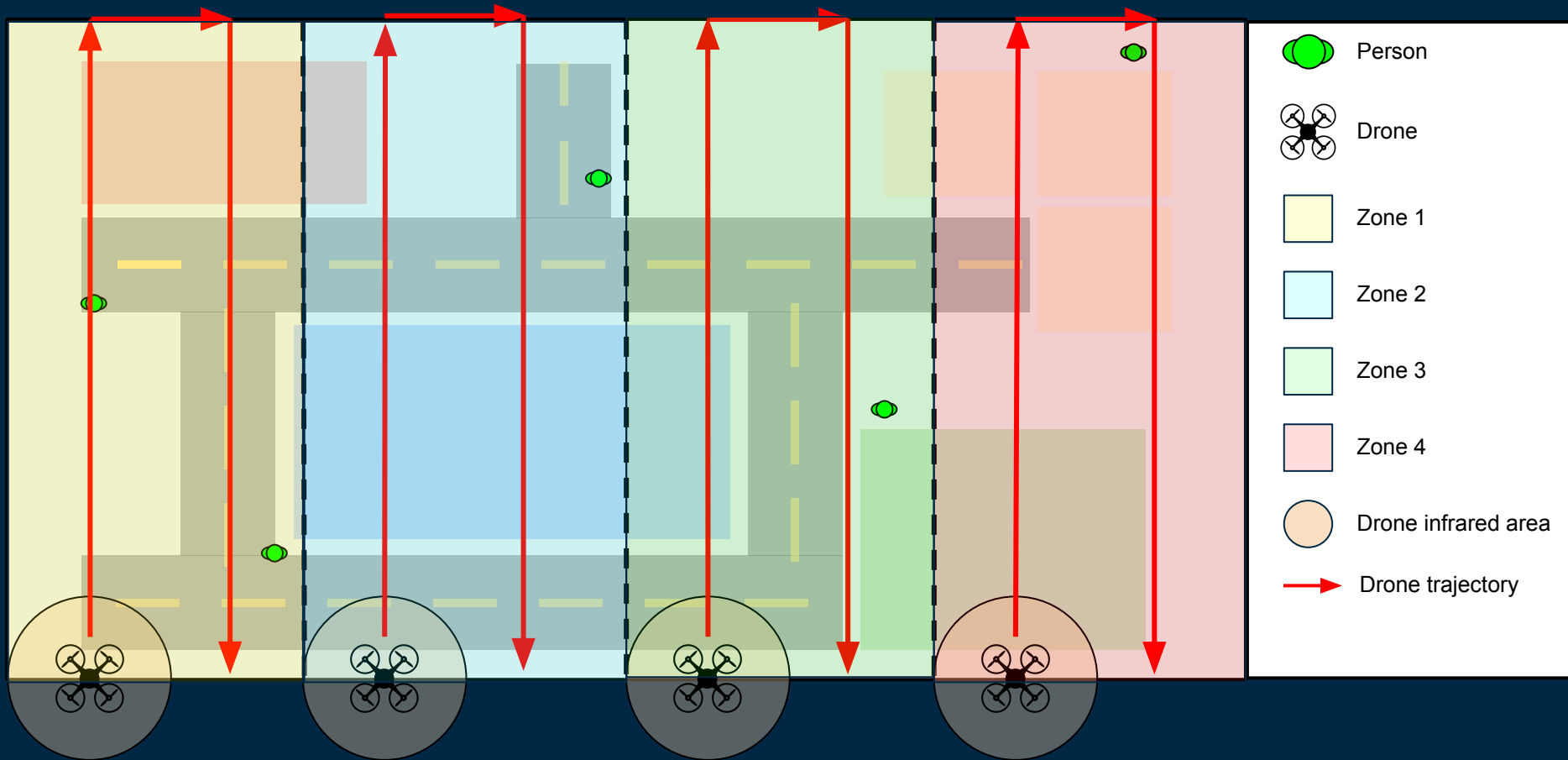
Le schéma ci-dessus représente l'environnement de la simulation. La stratégie pour la reconnaissance de personnes est divisé en deux partie principales. La reconnaissance primaire qui a comme but estimer à peu près la position et la reconnaissance de précision qui va diminuer l'incertitude de l'estimation pour envoyer aux Huskys.



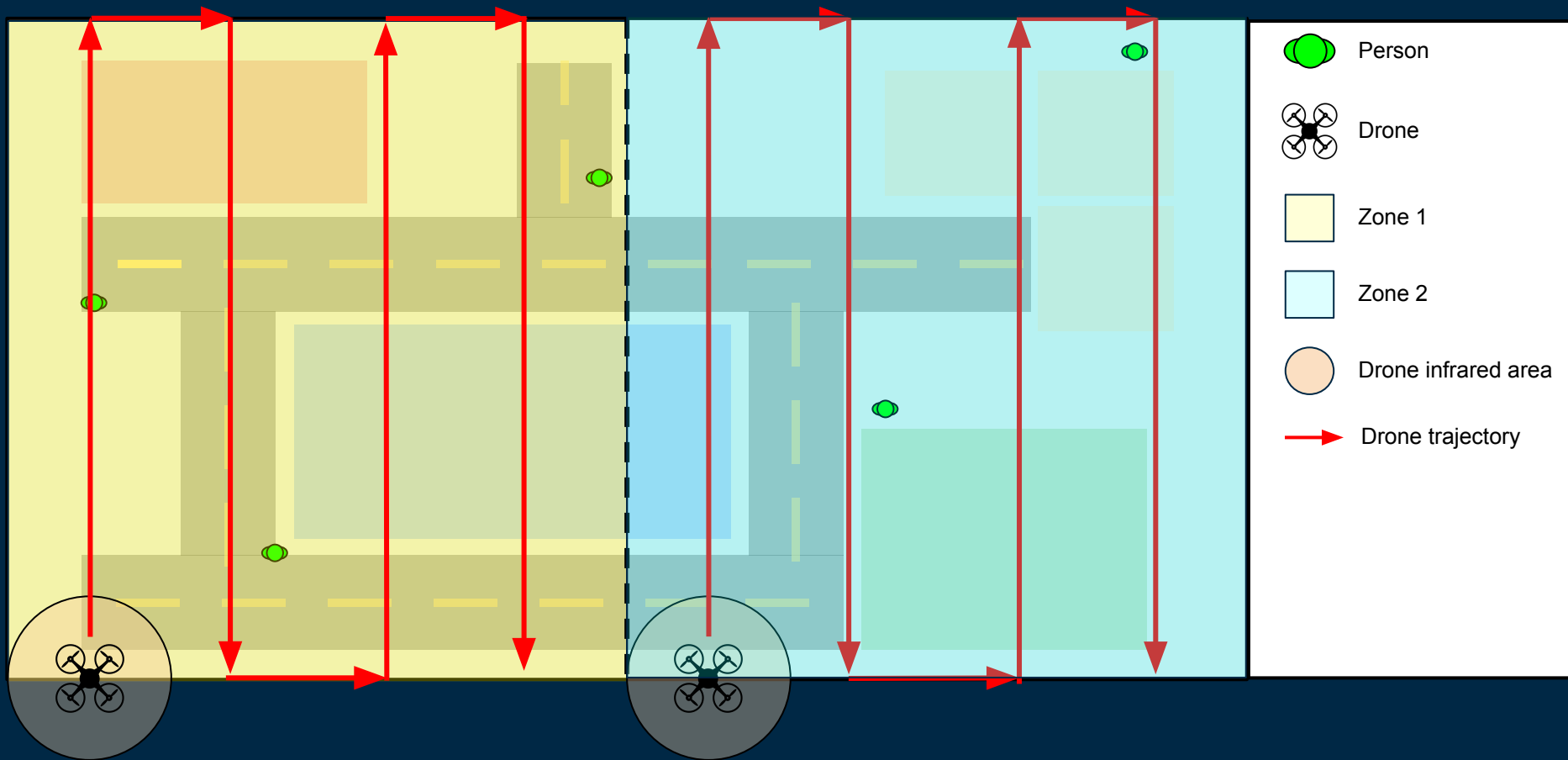
Reconnaissance Primaire



Pour la reconnaissance primaire, l'environnement est répartie en quatre zones et chacune des ces zones est attribué à un drone

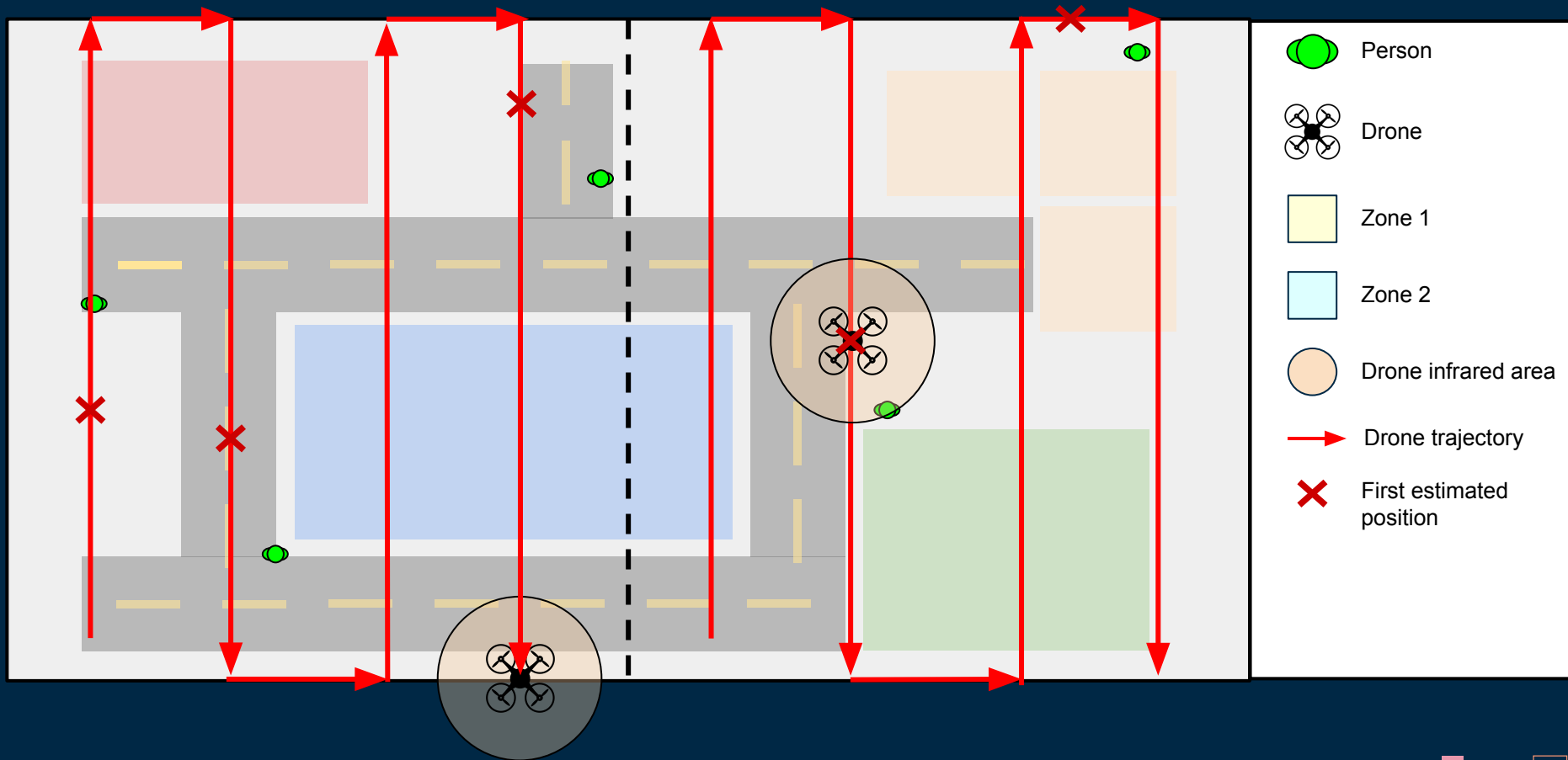


Les drones vont parcourir sa respective zone de la façon représentée ci-dessus et utilisent des capteurs infra-rouge pour détecter les personnes hostiles. De cette façon, au final de la trajectoire on aura des premières positions estimés pour les personnes

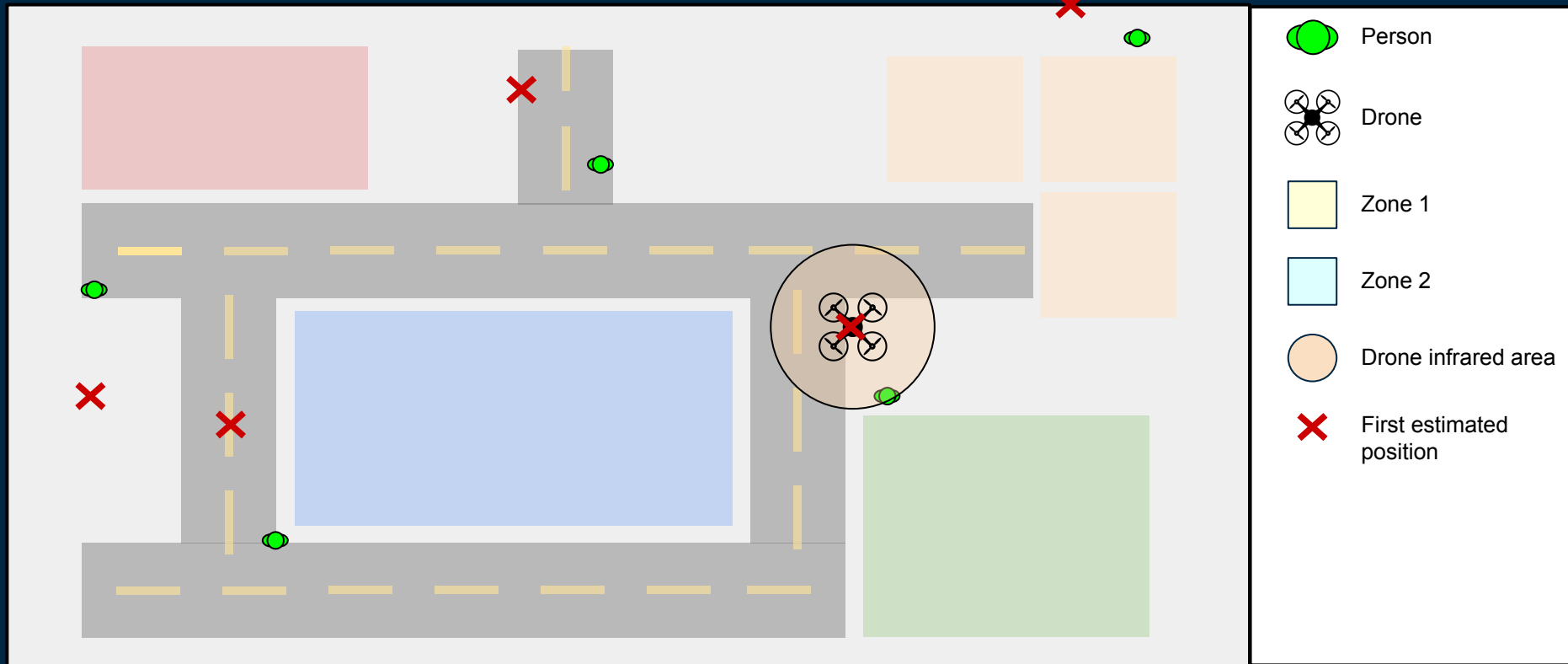


Le code `Env_strategy` est responsable pour la définition des goals intermédiaires et donc la trajectoire. Ici on montre le cas où on utilise seulement deux drones et le trajectoire calculée par l'algorithme

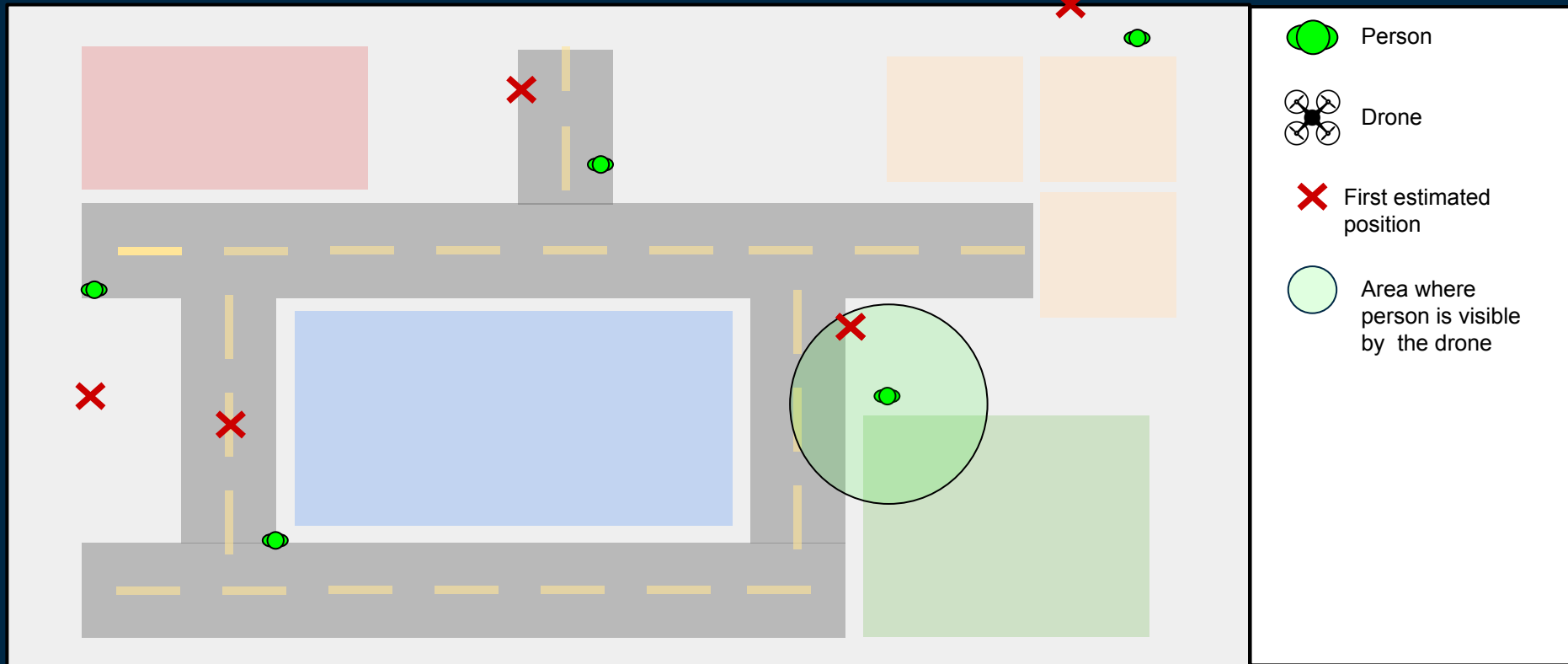
Reconnaissance de Précision



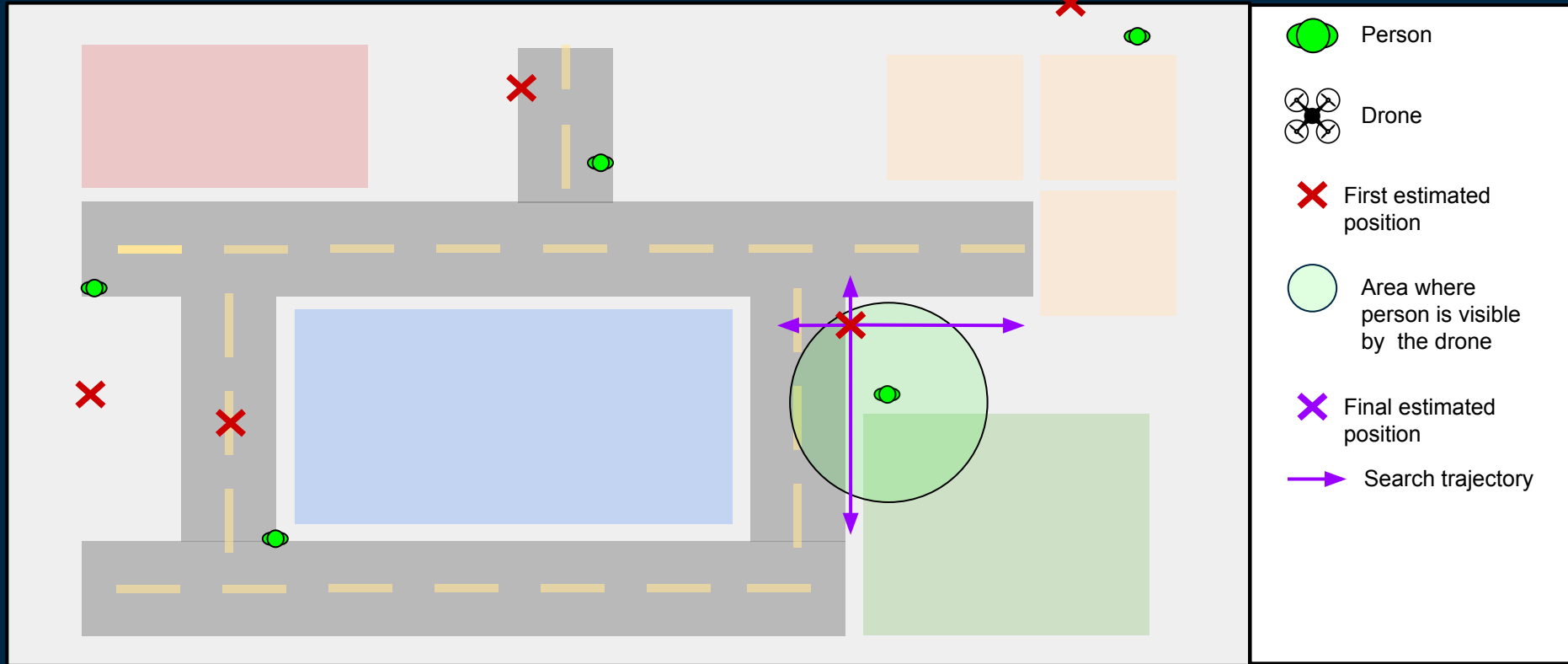
Dans le schéma ci-dessus on représente les premières estimations de la position après la reconnaissance primaire.



Le drone responsable pour la reconnaissance de précision se dirige vers la position estimé par le reconnaissance primaire.

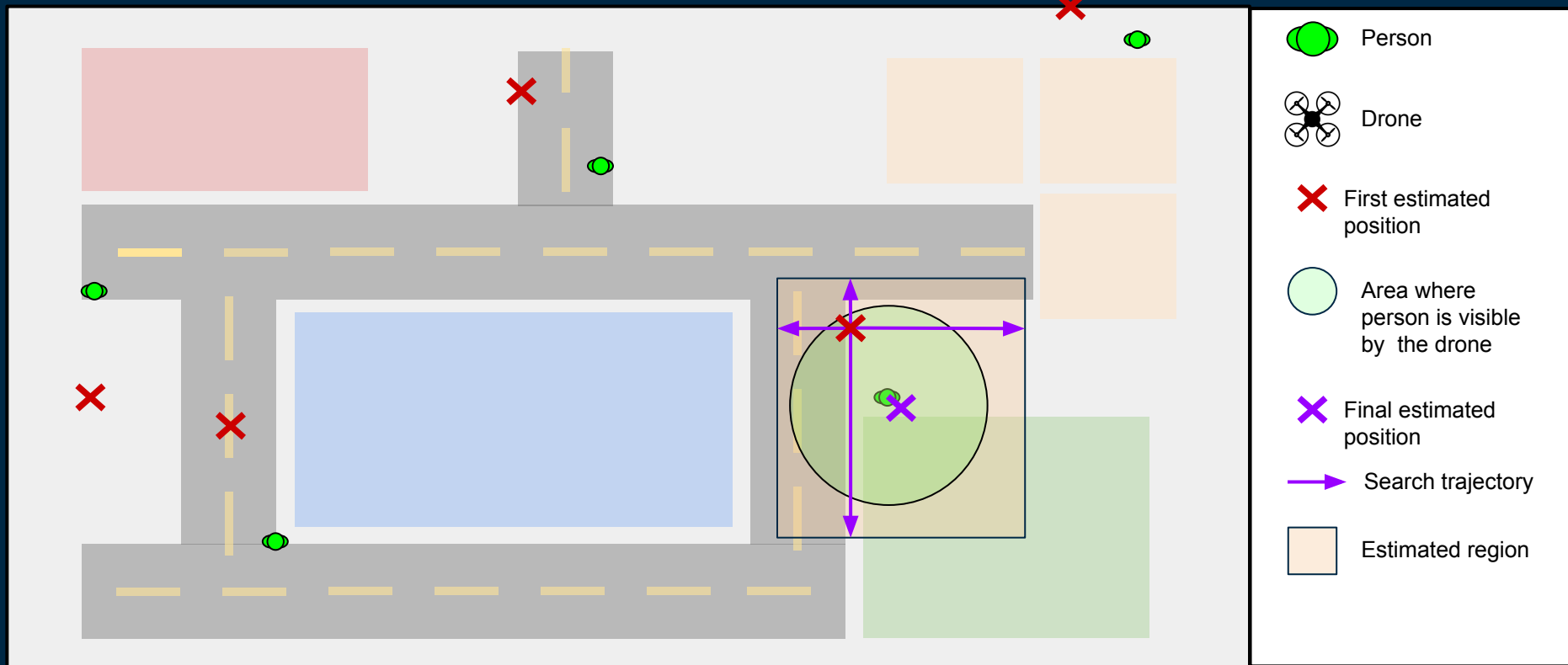


Le schéma ci-dessus montre les rayons dans lequel la personne est détecté par le capteur infra rouge.



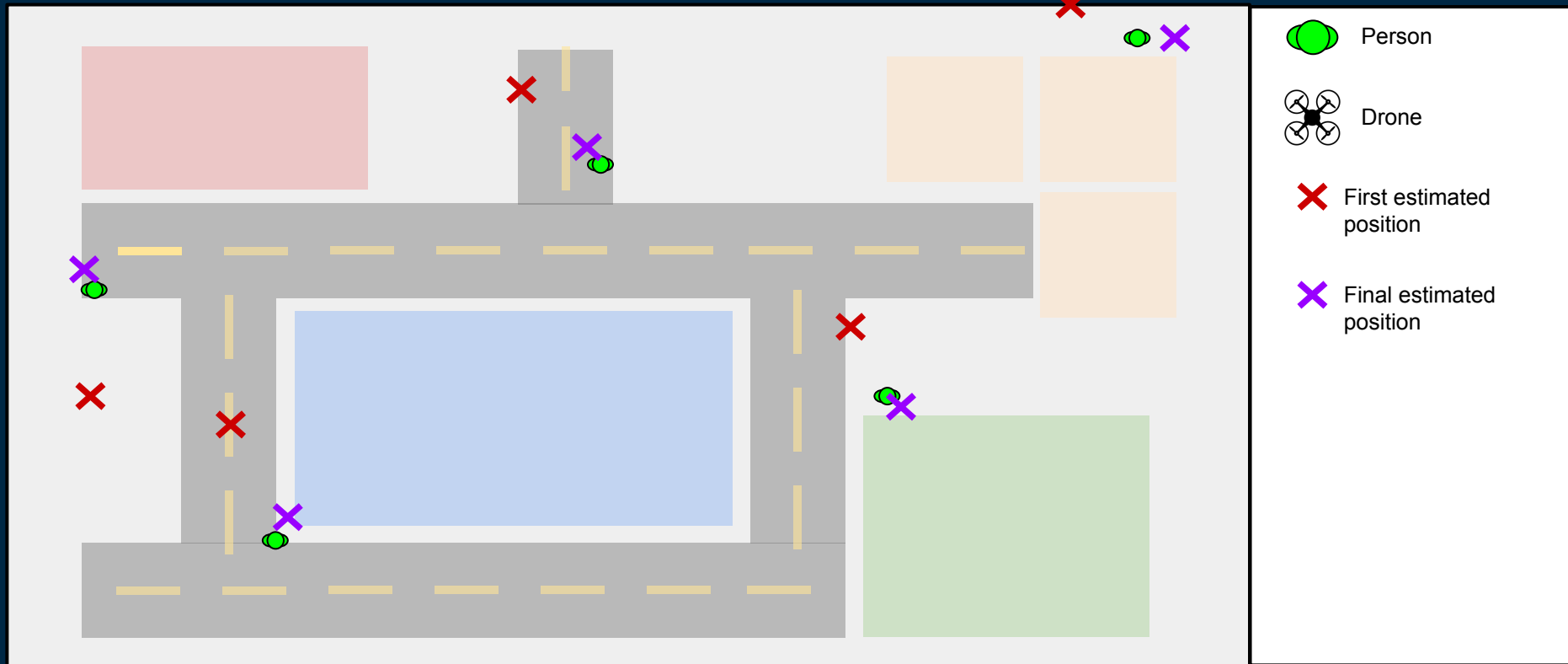
Ensuite, le drone responsable pour la reconnaissance de précision exécuté une trajectoire de recherche autour de la personne pour préciser son position. Cette trajectoire de recherche consiste dans le déplacement du drone dans les direction vertical et horizontal jusqu'à ce qu'il arrive plus à détecter la personne.





De cette façon, il est possible de faire une triangulation pour estimer la position de la personne.





Le schéma ci-dessus montre les positions initiales et les positions finales estimées. En utilisant ce méthode il est possible de fournir des position beaucoup plus précises des personnes aux Huskys.

DRONE - Résultats sans Killzone

	Hostile 1 (Drone 3)	Hostile 2 (Drone 4)	Hostile 3 (Drone 2)	Hostile 4 (Drone 3)	Hostile 4 (Drone 2)	Hostile 5	Hostile 6 (Drone 4)	Hostile 7 (Drone 4)	Hostile 8
Position réelle	X: -3.84 Y: -12.78	X: 21.72 Y: 41.11	X: -11.66 Y: 35.61	X: -7.40 Y: 33.55	X: -7.40 Y: 33.55	X: -33.12 Y: 50.56	X: 24.32 Y: -27.77	X: 30.82 Y: 18.74	X: 113.69 Y: 13.71
Position estimée	X: 0.30 Y: -20.31	X: 23.71 Y: 31.47	X: -19.85 Y: 27.56	X: -16.08 Y: 39.24	X: 0.88 Y: 24.49	Non trouvé	X: 20.37 Y: -36.82	X: 33.40 Y: 23.61	Non trouvé
Position précise	X: 0.30 Y: -13.31	X: 25.71 Y: 37.47	X: -11.85 Y: 32.56	X: -7.08 Y: 38.24	X: -4.11 Y: 33.49	Non trouvé	X: 24.37 Y: -27.82	X: 30.40 Y: 18.61	Non trouvé

DRONE - Résultats avec Killzone

	Hostile 1 (Drone 3)	Hostile 2 (Drone 4)	Hostile 3 (Drone 2)	Hostile 4 (Drone 3)	Hostile 4 (Drone 2)	Hostile 5	Hostile 6 (Drone 4)	Hostile 7 (Drone 4)	Hostile 8
Position réelle	X: -3.84 Y: -12.78	X: 21.72 Y: 41.11	X: -11.66 Y: 35.61	X: -7.40 Y: 33.55	X: -7.40 Y: 33.55	X: -33.12 Y: 50.56	X: 24.32 Y: -27.77	X: 30.82 Y: 18.74	X: 113.69 Y: 13.71
Position estimée	X: 1.28 Y: -27.06	X: 32.53 Y: 21.29	X: -20.40 Y: 28.41	X: 8.76 Y: 19.33	X: -17.64 Y: 40,31	Non trouvé	X: 18.35 Y: -45.69	X: 40.89 Y: 12.82	Non trouvé
Position précise	X: 3.72 Y: -13.06	X: 23.61 Y: 38.73	X: -11.40 Y: 35.41	X: -7.64 Y: 33.31	X: -5.92 Y: 31.47	Non trouvé	X: 24.35 Y: -27.69	X: 31.40 Y: 17.16	Non trouvé

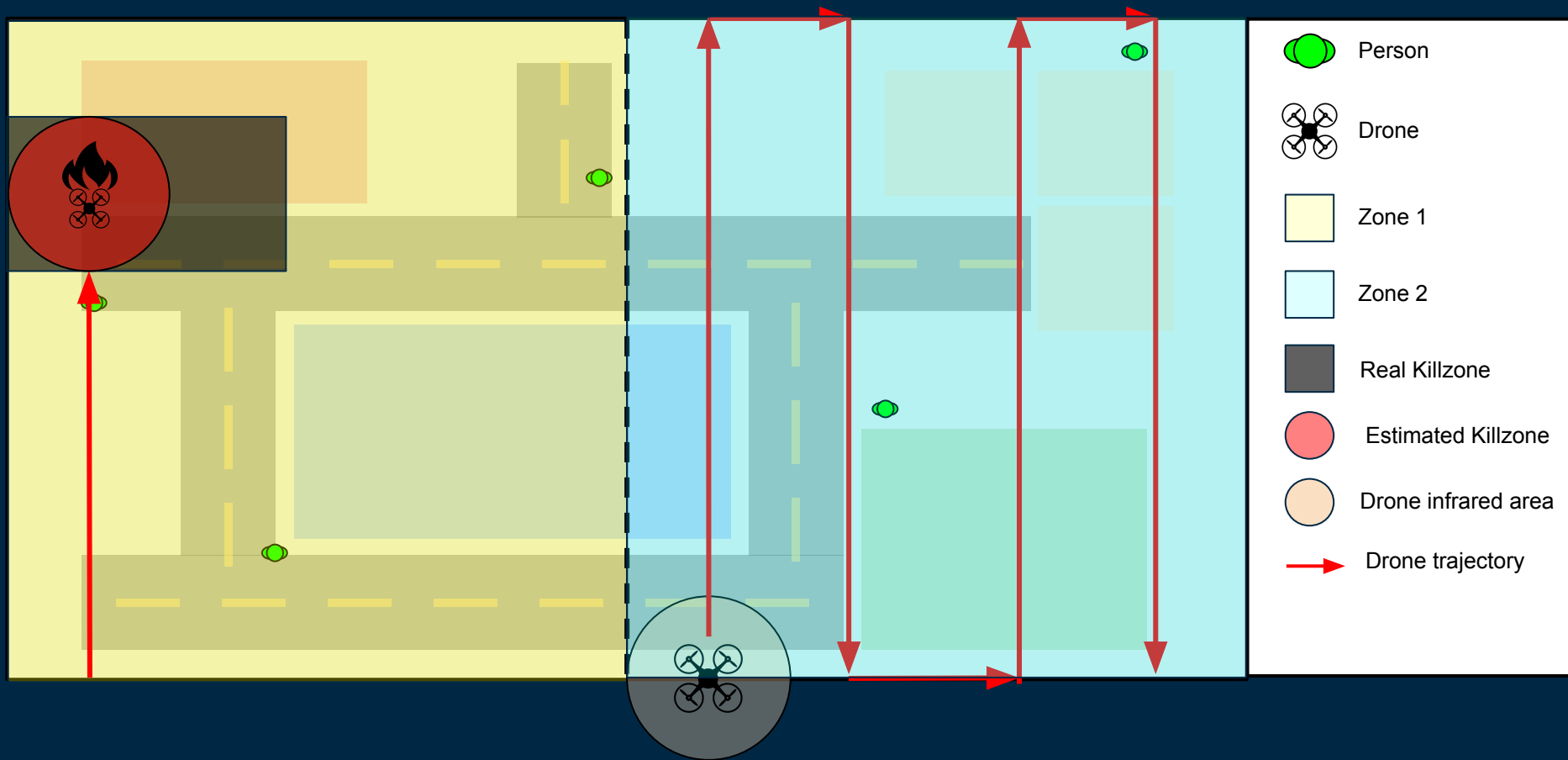
DRONE - Analyse des Résultats

Tout d'abord, on peut observer que la stratégie établie dépend de l'environnement. Cela est dû au fait que les drones n'utilisent pas de capteurs de profondeur pour éviter les obstacles. En outre, selon les positions de la Killzones, les résultats peuvent varier considérablement.

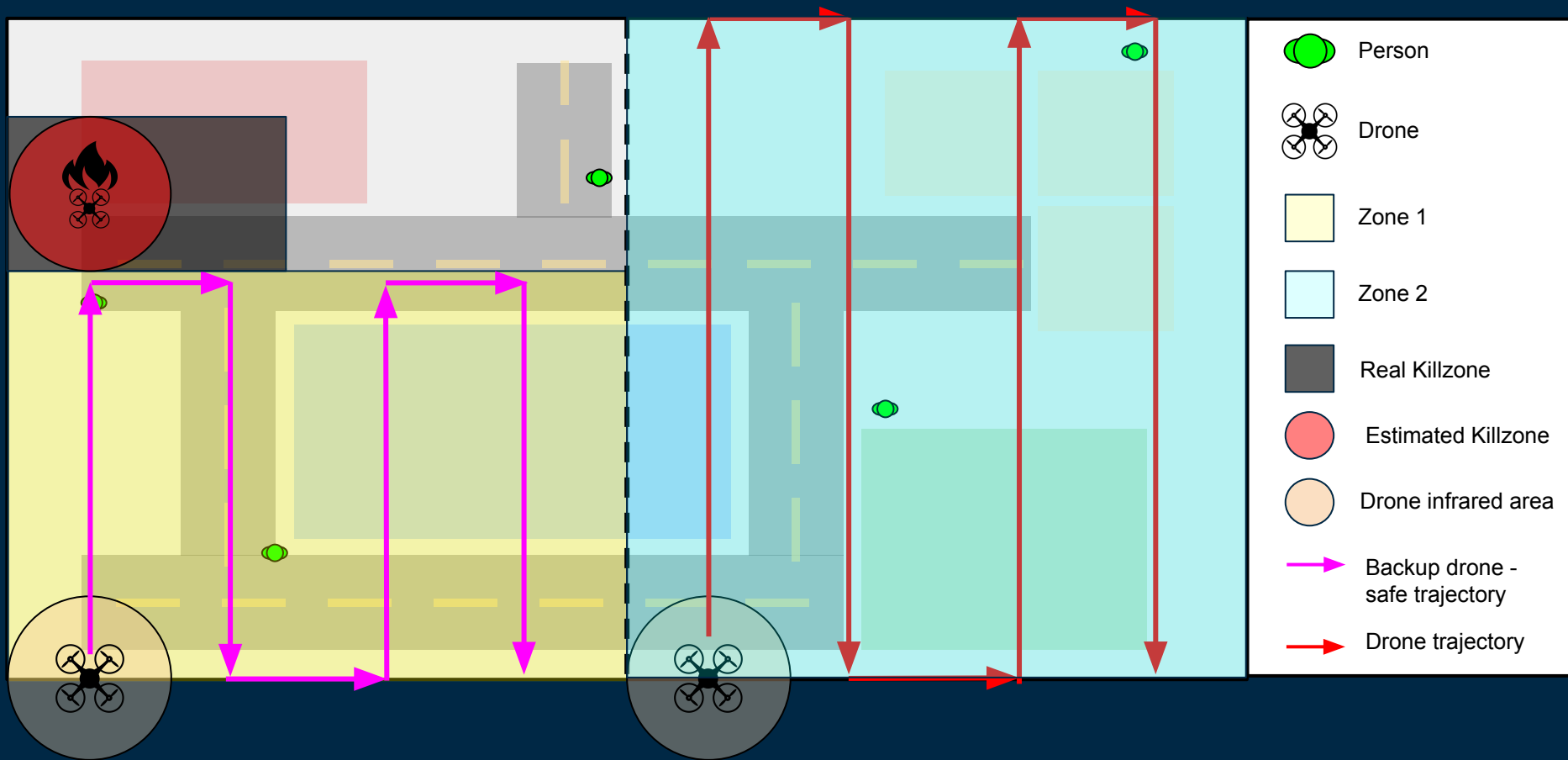
Néanmoins, pour l'environnement et les positions des Killzones considérés, les résultats finaux des positions des personnes hostiles étaient très proches des positions réelles. Cela valide la stratégie adoptée de la reconnaissance primaire, suivi par la reconnaissance de précision.

Toutefois, la détermination des positions prends beaucoup de temps.

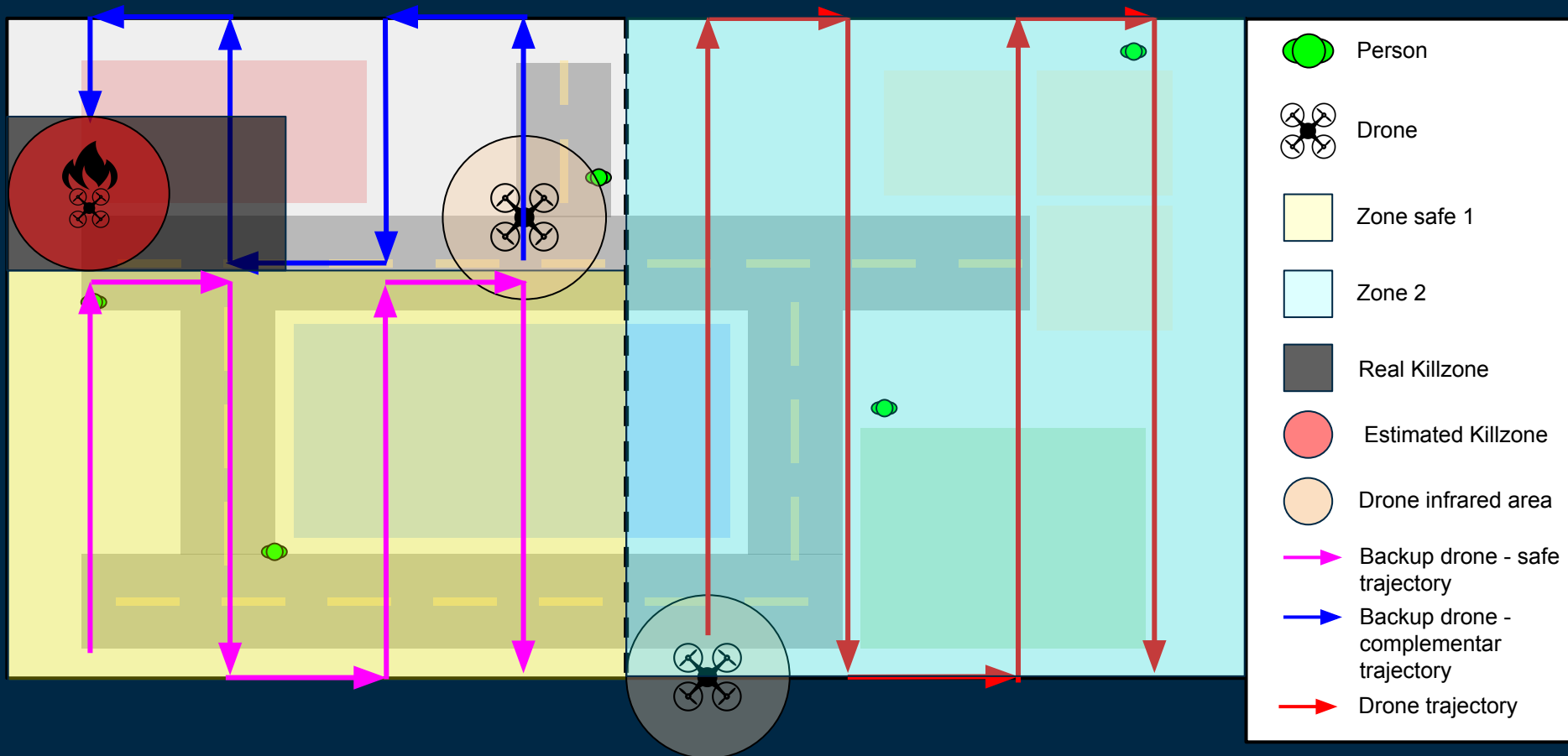
Stratégie pour la Killzone – non implémentée



Dans ce schéma on suppose qu'une drone a été abattu dans un killzone pendant son trajectoire de reconnaissance primaire. Une fois abattu, on sauvegarde la position où le drone a été abattu et on estime une zone que doit être évité.



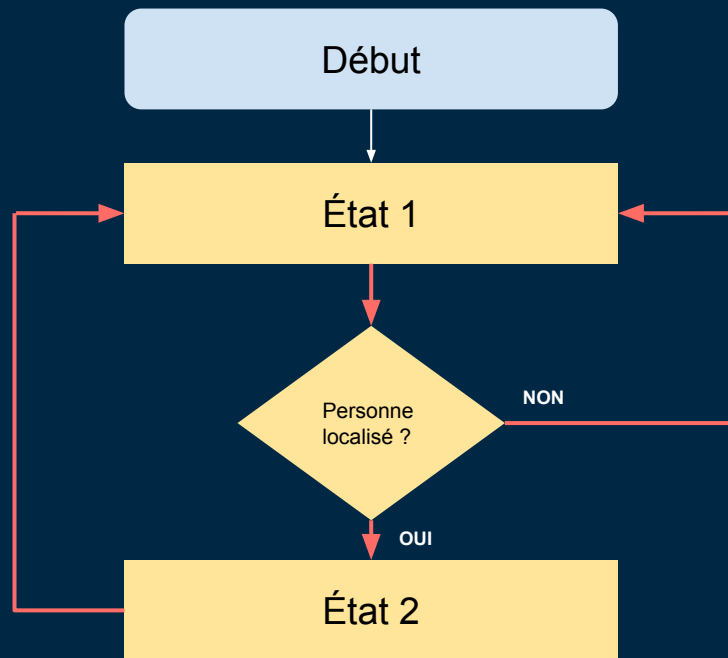
Ensuite, on utilise le backup drone pour continuer la reconnaissance primaire. On calcule une nouvelle trajectoire pour éviter la killzone. Le drone va rechercher des personnes seulement dans la nouvelle zone safe 1.



Finalement, après avoir fait la recherche dans la zone safe 1. Le backup drone va exécuter la trajectoire complementar indiqué par le chemin bleu. Dans le cas montré, le drone serait abattu, parce que la killzone est plus large que l'estimation fait.

COMMUNICATION

- Code responsable pour cette étape:
`Husky/machine_etat_husky.py`
- Machine d'état concernant les étapes d'actuation du Husky:
 - Où nous avons deux états.
 - Une étape de localization et une étape d'approche.

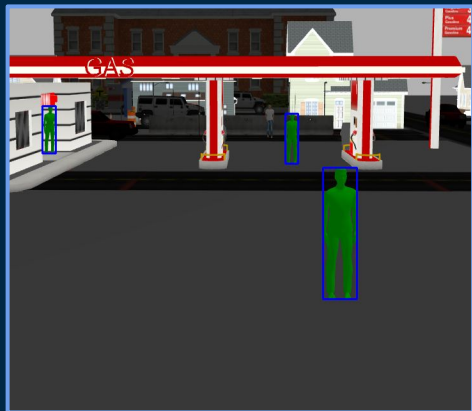
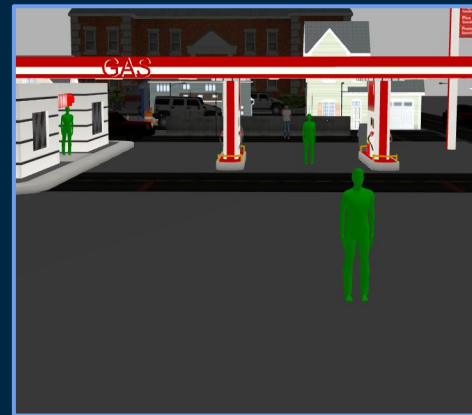


Communication - husky et machine d'état

- État 1:
 - Responsable de l'utilisation du "move_base" de Husky de manière "aléatoire", où Husky doit faire le tour de la carte, jusqu'au moment où il trouve une personne à suivre.
 - Il stocke ensuite ces données et passe à l'état suivant ;
- État 2:
 - Après avoir lu le vecteur qui contient les données, Husky atteindra la cible, en évitant les obstacles au milieu du parcours ;
 - A la fin, en s'approchant de la cible, il l'identifiera (par proximité);
 - Une fois identifié le cible, nous reviendrons à l'état initial.


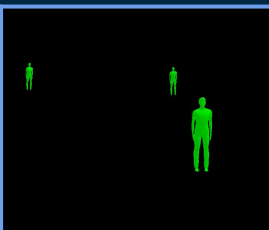
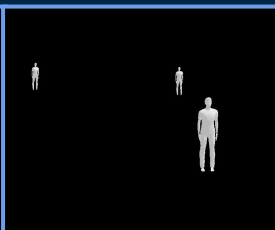
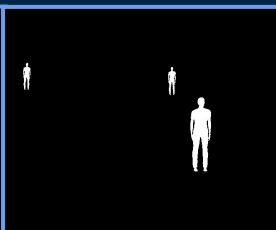
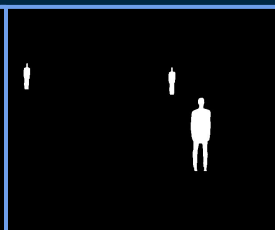
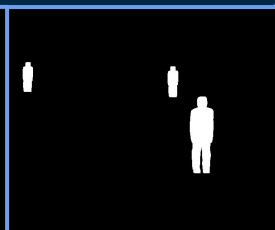
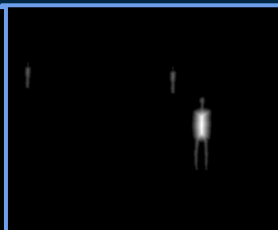
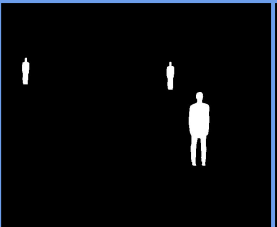
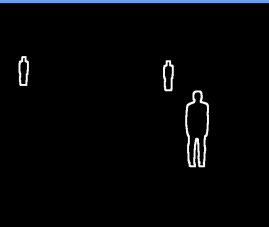
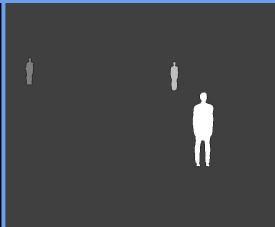
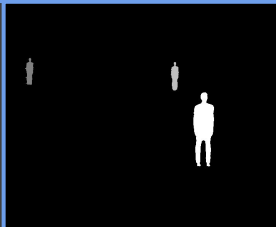
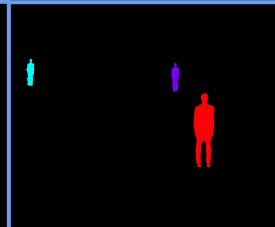

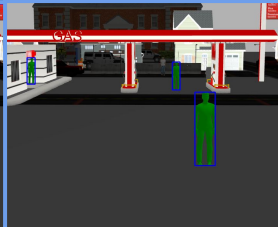
IDÉES ÉTUDIÉES MAIS NON IMPLÉMENTÉES – Détection des personnes

Afin de pouvoir détecter les personnes hostiles dans l'environnement, une méthode par le traitement des images obtenues par les caméras du drone et du husky a été étudiée



Dans cette méthode, en utilisant la couleur spécifique des personnes une série de traitements et de transformations ont été effectués sur l'image originale jusqu'à obtenir les régions de l'image où les personnes étaient

IDÉES ÉTUDIÉES MAIS NON IMPLÉMENTÉES – Détection des personnes

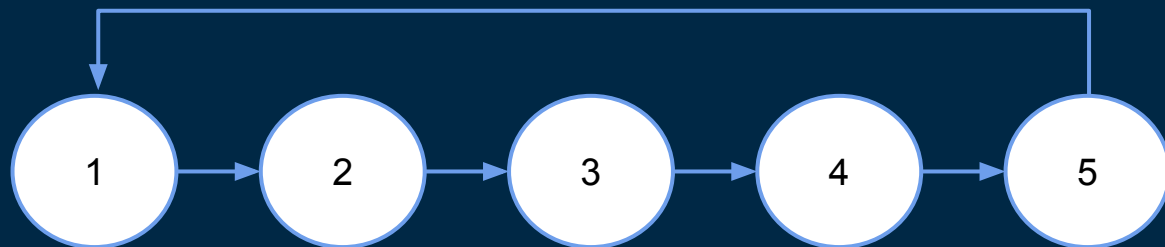
						
1. True frame	2. Color Threshold	3. Grayscale	4. Binary Threshold	5. Closing	6. Sure Background	7. Distance Transformation
Image prise par la caméra frontale du drone	Niveaux de gris de l'image avec seulement la couleur désirée	Image contenant uniquement les pixels du champ de couleur sélectionné	Seuil binaire de l'image en niveaux de gris	Transformation morphologique pour fermer les petits trous sur les personnes	Dilater l'image précédente pour être sûr d'obtenir les bords de chaque personne	Utiliser la transformation de la distance pour obtenir les régions dont on est sûr qu'il s'agit d'une personne
						
8. Sure Foreground	9. Unknown Area	10. Watershed	11. Labeled Image	12. Colored Labeled Image	13. Frame with contour	14. Frame with windows
En appliquant un seuil à partir de l'image 7, nous obtenons la zone dont on est sûr qu'il s'agit d'une personne	Soustraire l'image 8 de l'image 6 pour obtenir la zone dont on n'est pas sûr qu'il s'agisse d'une personne ou non. De cette façon on peut déterminer des personnes qui sont très proches l'une de l'autre	Utilisation des algorithmes Connected Components et Watershed pour identifier chaque personne et obtenir leurs contours	Image marquée sans l'arrière-plan de l'image précédente	Transformez les labels en niveaux de gris en labels RGB	Contour de chaque personne sur l'image originale	Des fenêtres qui contiennent la région entière de chaque personne 32

IDÉES ÉTUDIÉES MAIS NON IMPLÉMENTÉES -

Estimation de la position des personnes par reconnaissance d'images

Une machine d'état a été développée pour estimer la position des personnes par reconnaissance d'image.

1. Stratégie de recherche de personnes
2. Une fois que le drone trouve une ou plusieurs personnes grâce à sa caméra, il tourne jusqu'à ce que la personne la plus proche (la plus grande fenêtre) soit au centre de l'image
3. Il s'élève ensuite jusqu'à une hauteur de 9 mètres (hauteur à laquelle il est possible de détecter des personnes avec le capteur infrarouge)
4. Avance linéairement jusqu'à ce que le capteur infrarouge détecte une personne
5. Il envoie sa position au robot terrestre



IDÉES ÉTUDIÉES MAIS NON IMPLÉMENTÉES -

Estimation de la position des personnes par reconnaissance d'images

Malheureusement, nous avons réalisé que cette méthode prendrait trop de temps et rencontrerait des difficultés avec des personnes complètement ou partiellement obstruées. Néanmoins, le code est utilisé pour la détection des personnes par le robot terrestre lorsqu'il se trouve à proximité d'une personne.



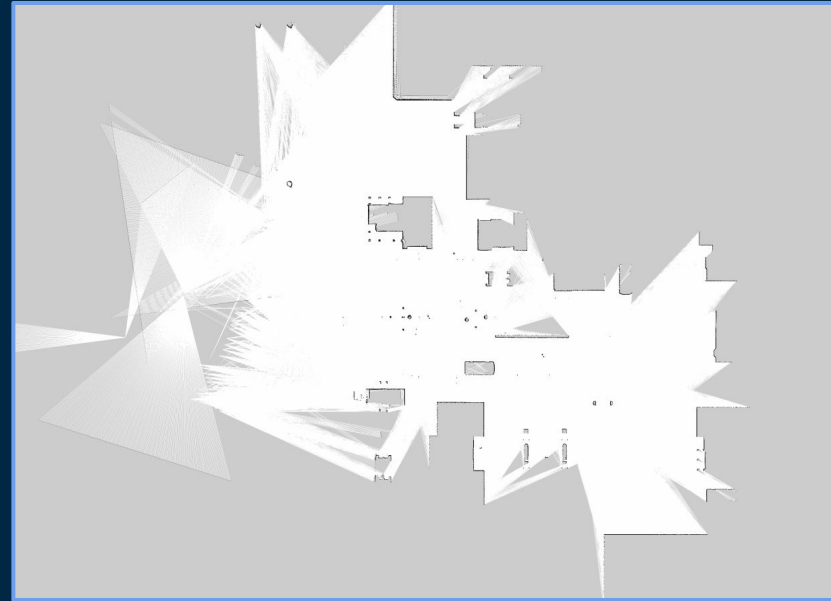
IDÉES ÉTUDIÉES MAIS NON IMPLÉMENTÉES –

Cartographie 2D

D'abord, nous avons étudié la possibilité de faire la cartographie de la simulation avec les drones. Pour ce faire nous avons évalué le niveau de performance des paquets de ROS "hector_mapping" et "gmapping".

Dans un premier instant, nous avons essayé "gmapping avec le rover et nous avons obtenu la carte de la diapositive.

Étant donné le bon résultat, nous avons décidé d'étudier leur implémentation sur les drones.

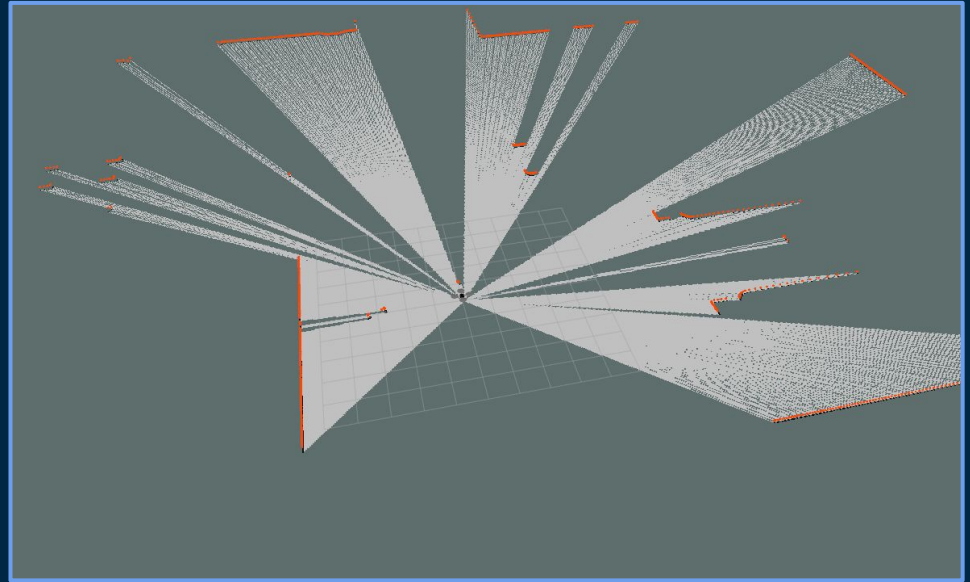


Carte obtenue avec "gmapping" en utilisant le husky



IDÉES ÉTUDIÉES MAIS NON IMPLÉMENTÉES – Cartographie 2D

Nous avons utilisé finalement le paquet “hector_mapping”, mais finalement nous avons trouvé plusieurs problèmes qu’on montrera dans la diapositive suivante.

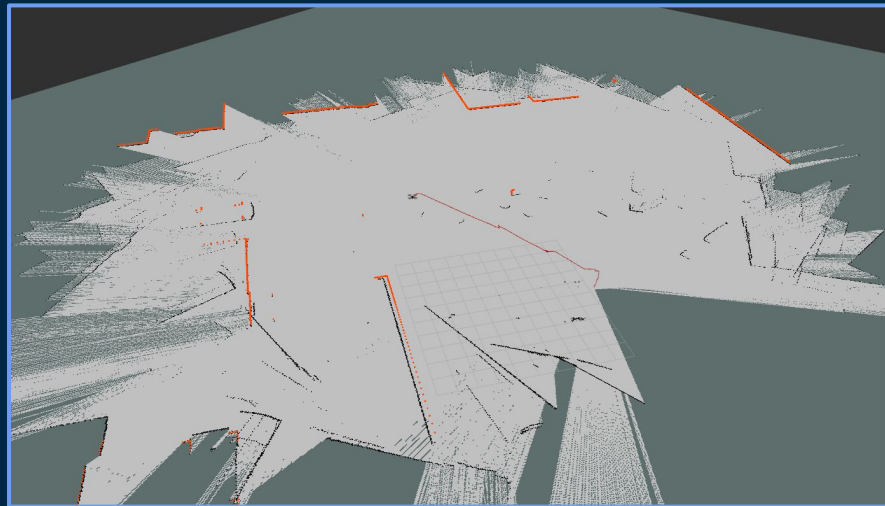


Carte obtenue avec “hector_mapping” en utilisant un drone dans un premier, sans décoller.

IDÉES ÉTUDIÉES MAIS NON IMPLÉMENTÉES – Cartographie 2D

Problèmes qui empêchent de mettre en oeuvre la cartographie 2D:

- La dynamique du drone lui fait s'incliner vers l'avant et le drone interprète le sol comme un obstacle.
- Problèmes pour coller la carte créée par chaque drone (paquet "multirobot_map_merge").
- Il faut voler suffisamment bas pour que la carte soit valable pour le rover.



Carte obtenue avec "hector_mapping" en utilisant un drone dans un premier, après avoir décollé et avoir avancé quelques mètres.

Travail futur

Améliorations pour notre code avec de code dont on dispose déjà ou de code qu'on pourrait développer:

- Machine d'états du husky - ajouter reconnaissance d'images développée pour s'approcher à la personne et un nouvel état qui lui permet de faire cela quand il est suffisamment proche de la position fournie par le drone.
- Drones: définir une stratégie de réorganisation pour la flotte de drones quand on perd un drone dans une zone hostile.
- Drones: définir une réorganisation de tâches pour les drones de la reconnaissance primaire. Pas de drones sans fonction.
- Améliorer la communication drone husky, pour avoir une liste globale de cibles (ne pas aller deux huskies vers la même personne ou un husky deux fois vers la même personne) et pour que le husky le plus proche au cible puisse agir en conséquence.
- Cartographie: nous recommandons un système 3D à cause des problèmes trouvés avec la cartographie 2D.



Merci