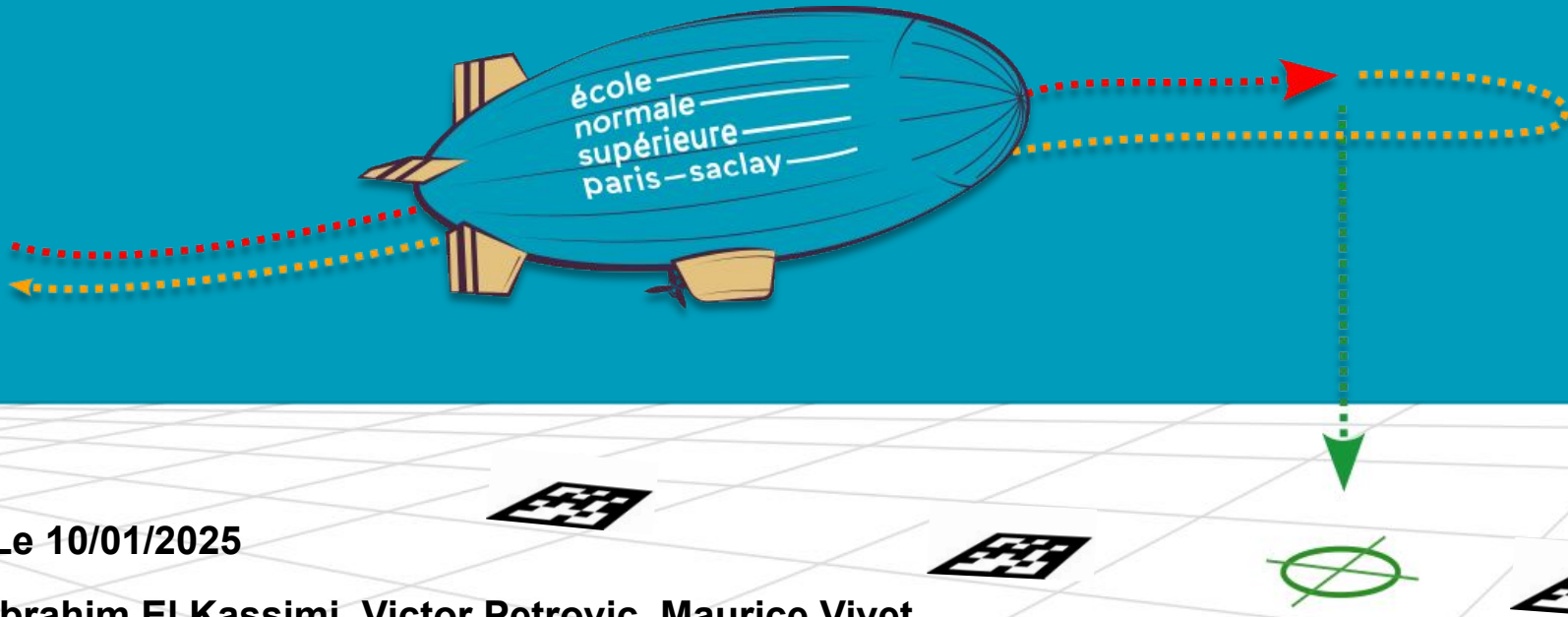


Méthodes de localisation par Apriltag à l'usage d'un dirigeable autonome

Projet CoBRA



Le 10/01/2025

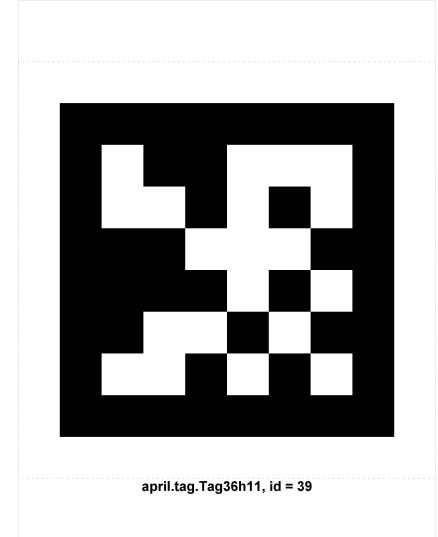
Ibrahim El Kassimi, Victor Petrovic, Maurice Vivet

- Localisation par Apriltag

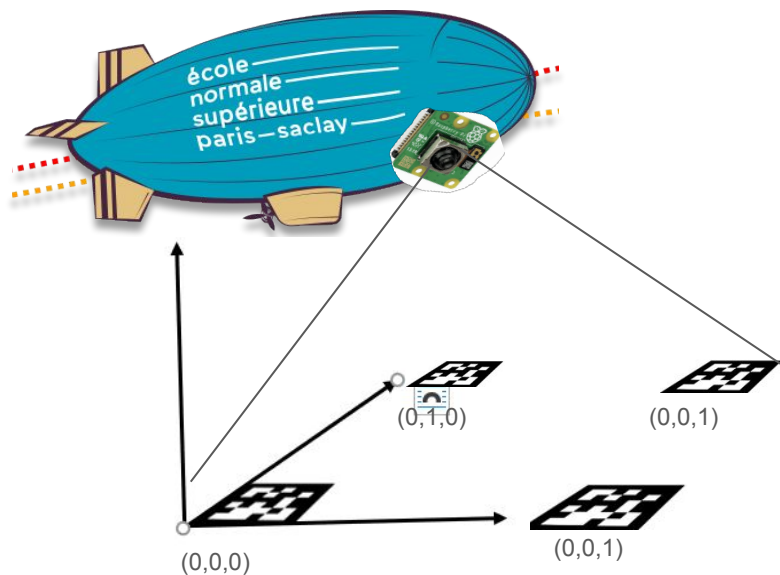
1. Principe

2. Théorie

3. Choix d'une configuration



1. Principe



acquisition de l'image

détection des tags

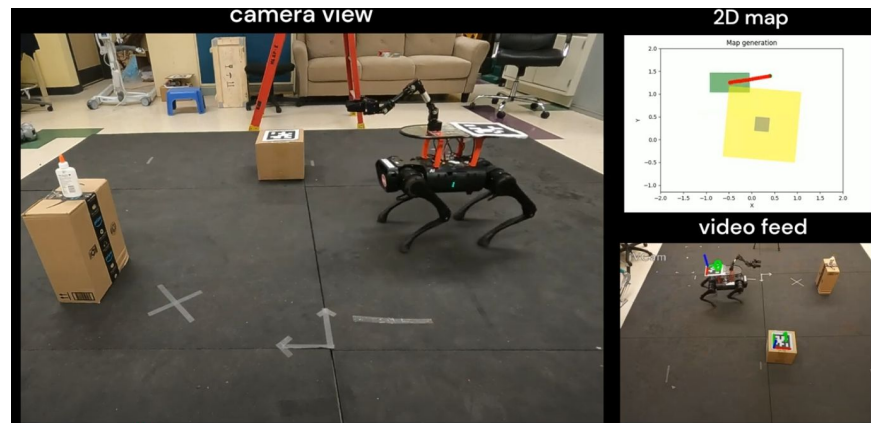
Calcul de la position du dirigeable

Utilisation

Apriltags:



<https://april.eecs.umich.edu/software/apriltag>



amazon

Matériel à disposition



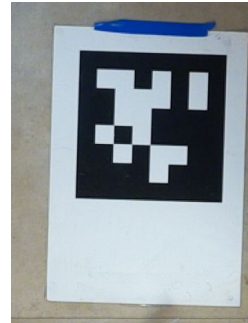
Nano-ordinateurs :

- Raspberry Pi Zero 2 W
- Raspberry Pi 5

Caméras :

- Pi Camera module V2
- Pi Camera module V3
- Pi Camera module V3 Wide

Feuilles A4 avec **tags** :



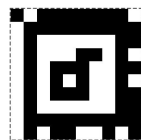
Code



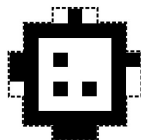
Tag36h11



TagStandard41h12



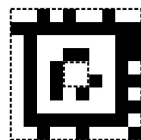
TagStandard52h13



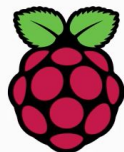
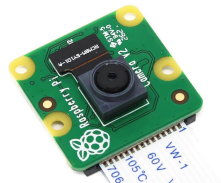
TagCircle21h7



TagCircle49h12



TagCustom48h12



Raspberry Pi OS



Système d'exploitation :

- Raspberry Pi OS

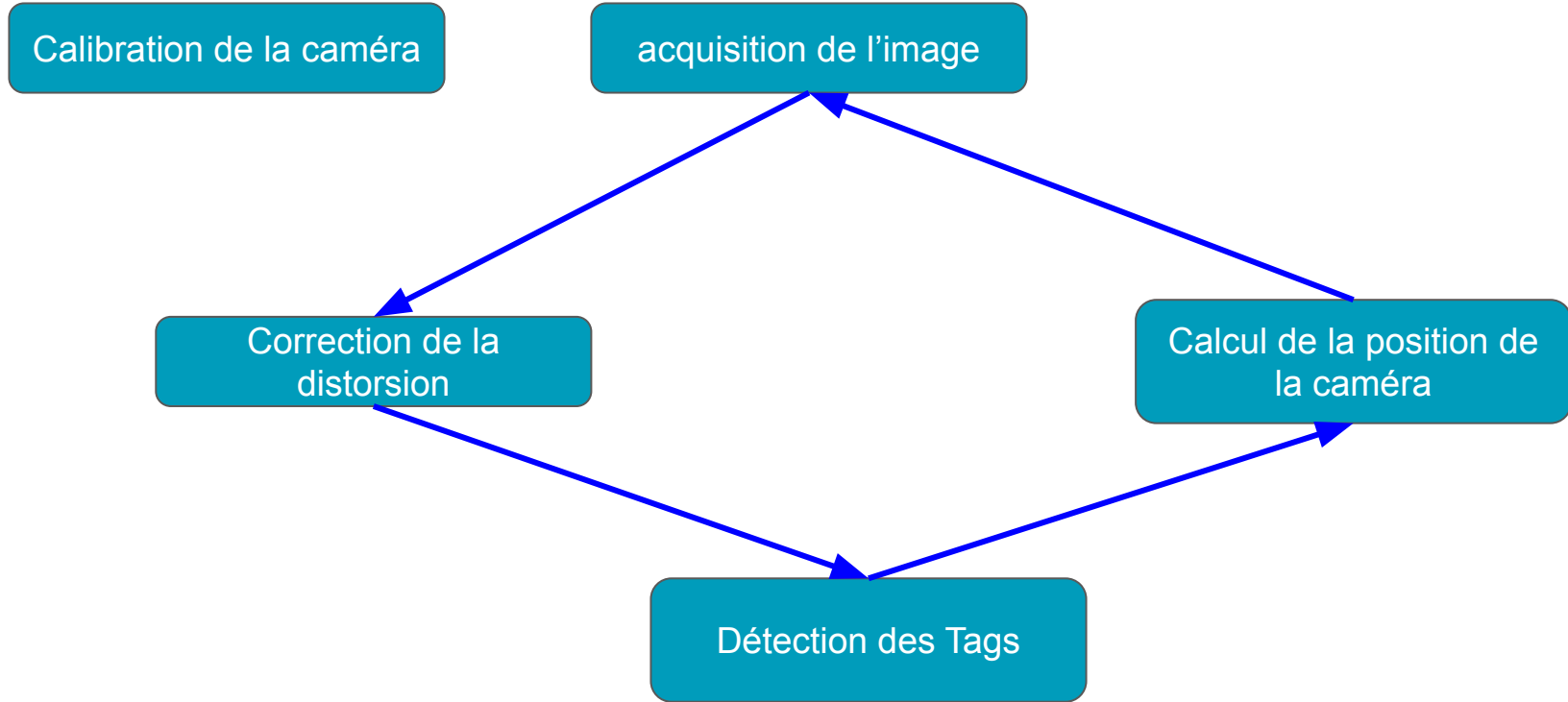
Bibliothèques utilisées :

- Récupération des images :
Picamera2
- Détection des tags :
dt-apriltags
- Traitement d'image :
OpenCV
- Calcul matriciel :
Numpy

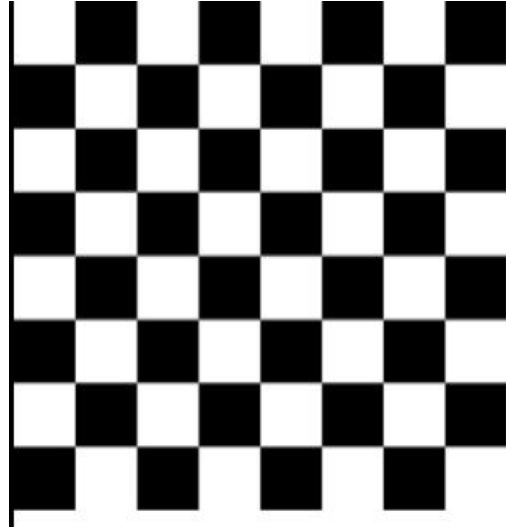
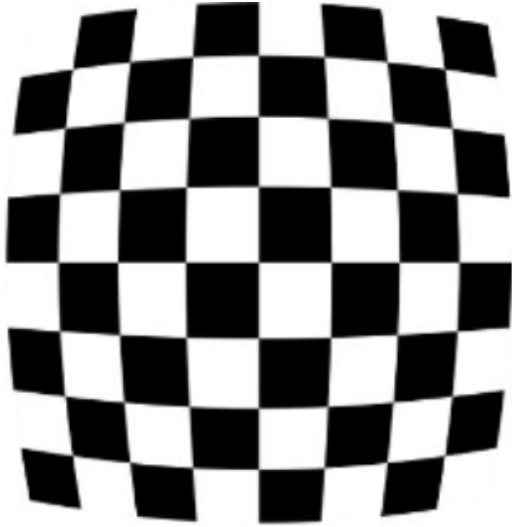
Avantages et inconvénients

Avantages	Inconvénients
<ul style="list-style-type: none">- Relativement économique- Précision suffisante (quelques cm)- Tags simples à installer	<ul style="list-style-type: none">- Sensible à luminosité- Dépend du champ de vision de la caméra- Nécessite l'installation de tags sur le sol

2. Théorie



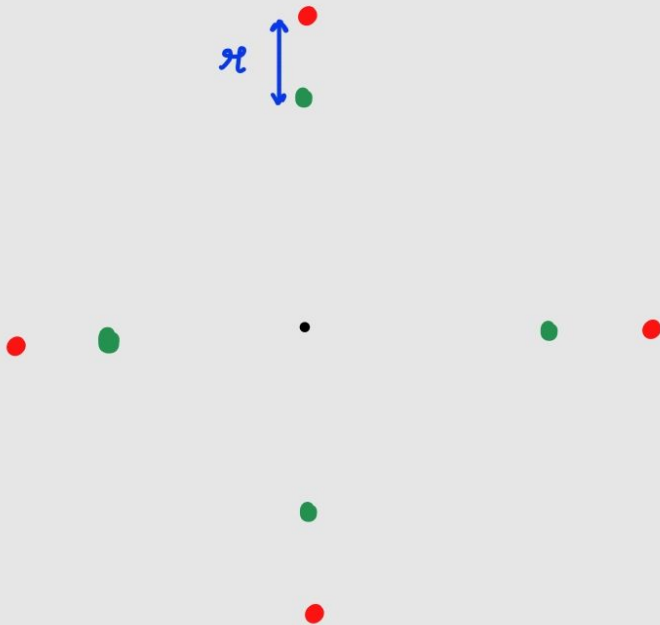
Calibration de la caméra



Distorsion Radiale

$$x_D = x \left(1 + K_1 r^2 + K_2 r^4 + K_3 r^6 \right)$$

$$y_D = y \left(1 + K_1 r^2 + K_2 r^4 + K_3 r^6 \right)$$



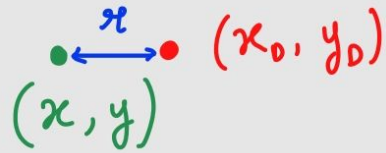
● P.image distordue

● P.image rectifiée

Distorsion Tangentielle

$$x_0 = x + \left[2p_1 xy + p_2 (x^2 + 2x^2) \right]$$

$$y_0 = y + \left[p_1 (x^2 + 2y^2) + 2p_2 xy \right]$$



● P.image distordue

● P.image rectifiée



echiquier1.jpg



echiquier2.jpg



echiquier3.jpg



echiquier4.jpg



echiquier5.jpg



echiquier6.jpg



echiquier7.jpg



echiquier8.jpg



echiquier9.jpg



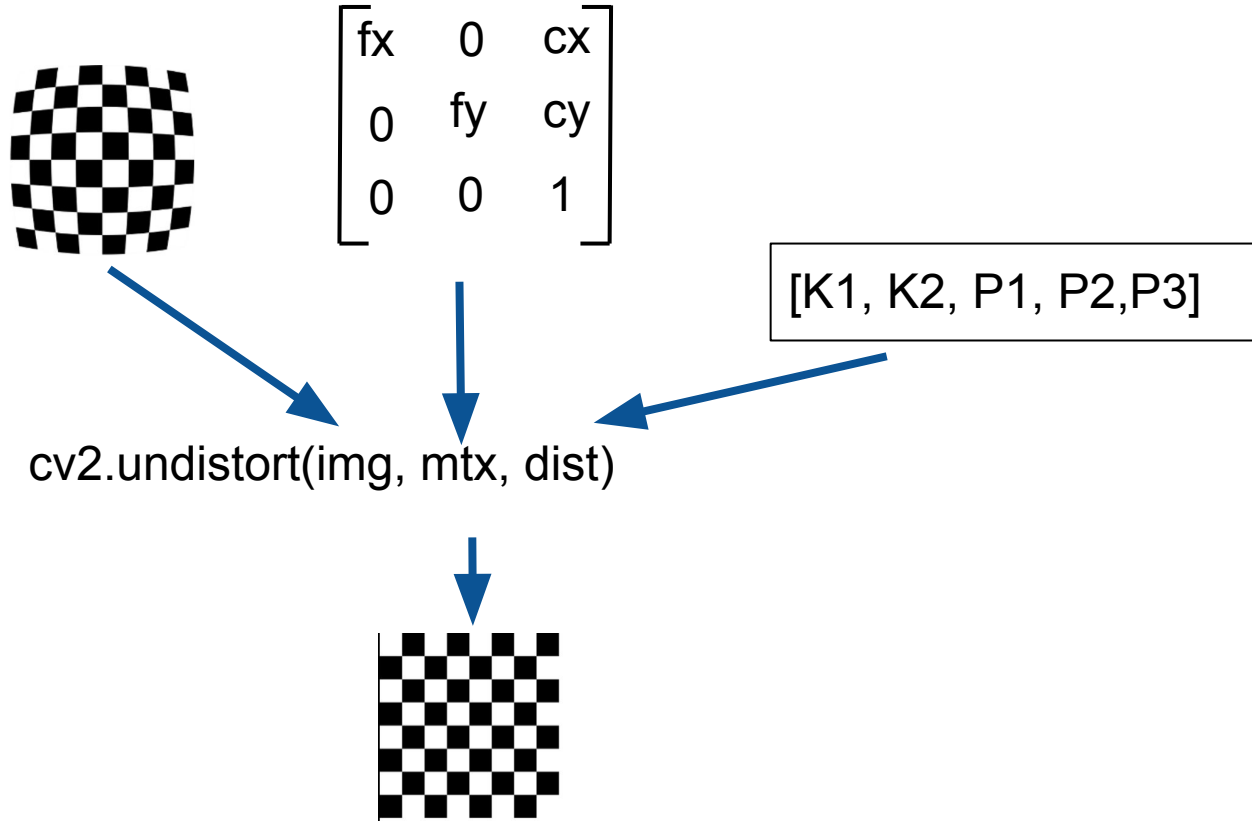
echiquier10.jpg



[K1, K2, P1, P2, P3]

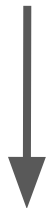
$$\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Rectification de l'image

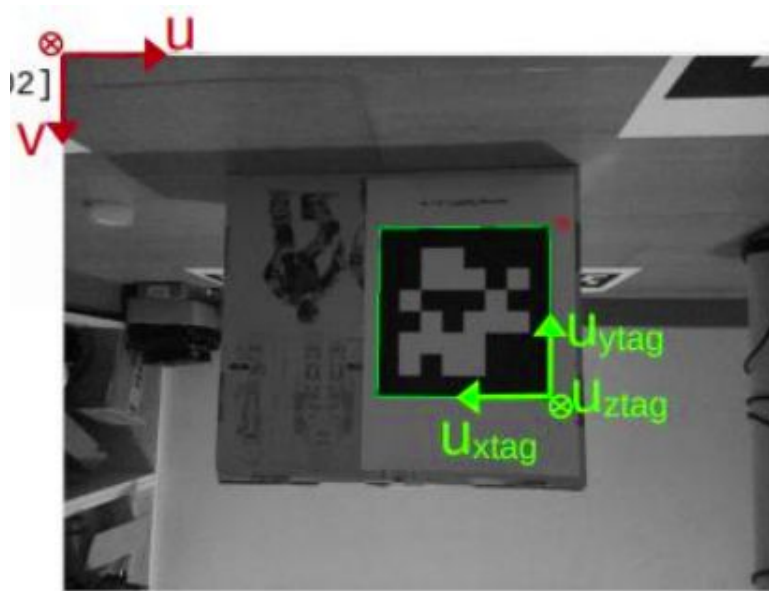


Détection du tag et algorithme pour la localisation

`Detector.detect(img_undistorted)`



tag_id : identifiant du tag
center : coordonnées du centre du tag
pose_R : (matrice 3x3)
pose_t : (vecteur)



Algorithme pour la localisation

pose_R (matrice 3x3)

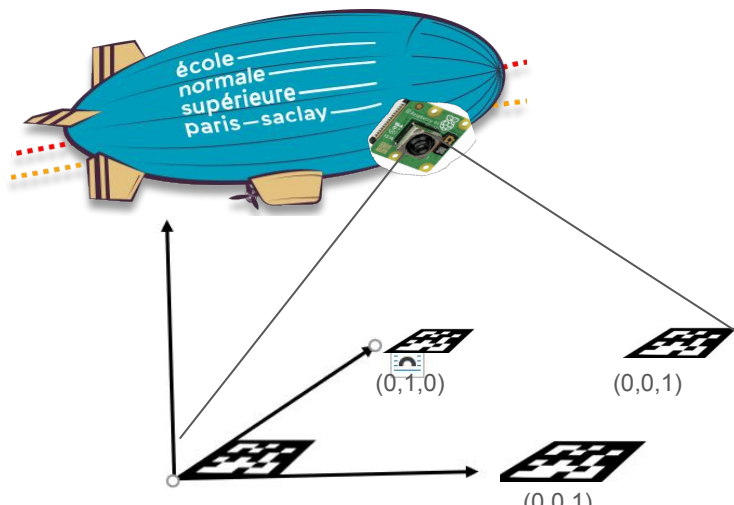
pose_t (vecteur)

matrice intrinsèque

position du tag par rapport au sol



Position de la caméra/sol
Son orientation



3. Choix d'une configuration

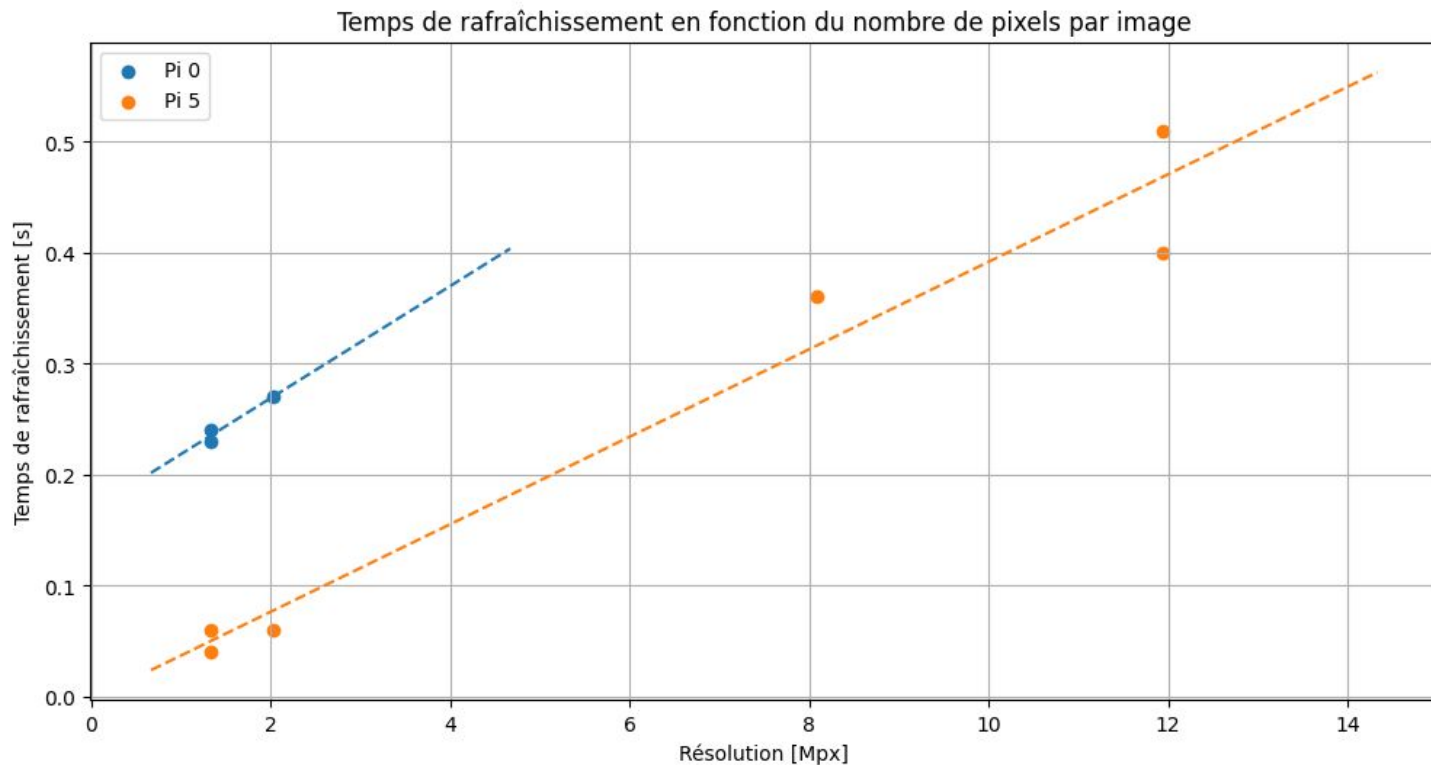
- Différentes mesures pour choisir la bonne configuration.
- A notre disposition : Deux nano-ordinateurs et 3 caméras.
- Mesures à une distance fixe d'un tag :
 - Précision de position et d'angle : fil à plomb et mètre ruban
 - Temps de rafraîchissement : chronomètre implémenté dans le code
 - Consommation électrique : multimètre USB
- Environnement de test : Lumière, distance, configuration... identiques pour assurer la répétabilité de l'expérience.

Précision et temps de rafraîchissement

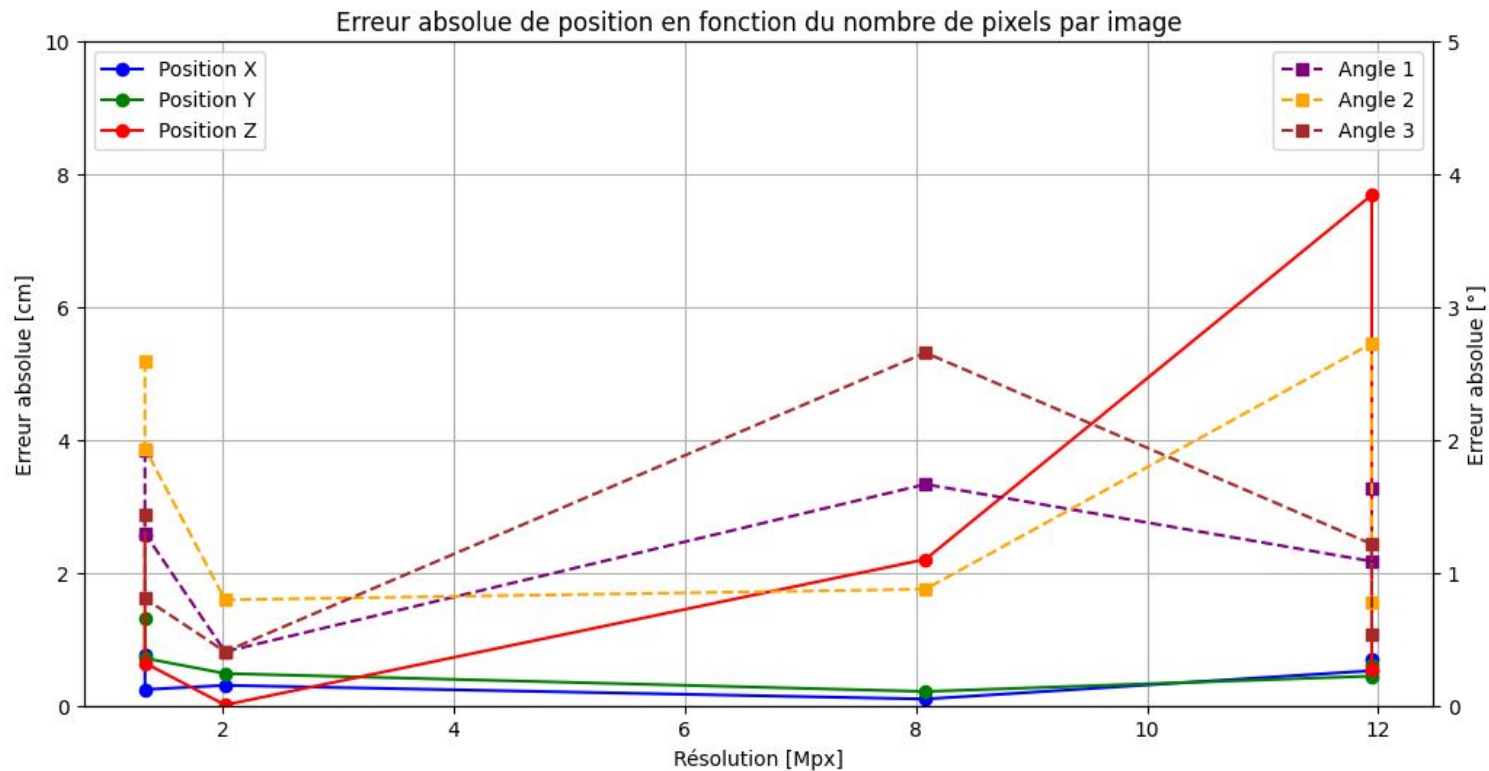
Caméra : Position (0,0,1275) et Angle (0,0,0) | Tag : Position (0,0,0) et Angle (0,0,0)

Résolution	Pi Camera V2				Pi Camera V3				Pi Camera V3 Wide			
	Temps de rafraîchissement (s)		Position xyz (mm)	Angle (°)	Temps de rafraîchissement (s)		Position xyz (mm)	Angle (°)	Temps de rafraîchissement (s)		Position xyz (mm)	Angle (°)
	Pi 0	Pi 5			Pi 0	Pi 5			Pi 0	Pi 5		
640 x 480 (recadrage)	Recadrage trop important											
1536 x 864 (recadrage)					0.23	0.06	(7.67, 13.28, 1300.74)	(1.92, -2.60, -1.44)	0.24	0.04	(2.50, -7.21, 1281.49)	(1.30, 1.94, 0.81)
1640 x 1232	0.27	0.06	(-3.13, 4.91, 1274.82)	(-0.41, -0.80, -0.41)								
1920 x 1080 (recadrage)	Recadrage trop important											
3280 x 2464	Mémoire insuffisante	0.36	(-1.07, 2.20, 1252,90)	(1.67, -0.88, -2.66)								
4608 x 2592					Mémoire insuffisante	0.51	(5.33 , -4.52, 1198.05)	(1.09, -2.73, 1.22)	Mémoire insuffisante	0.40	(-7.14, -6.21, 1280.54)	(1.64, -0.78, -0.54)

Temps de rafraîchissement



Précision

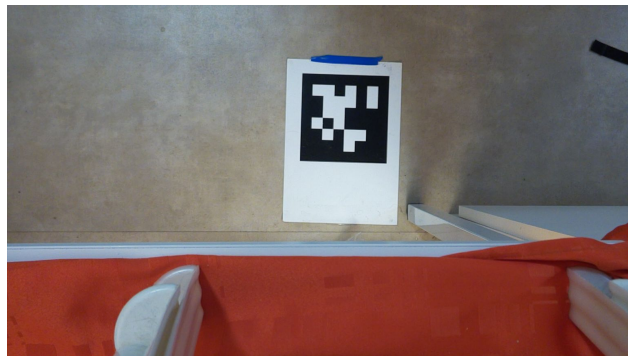


Choix d'une résolution

- Résolutions élevées : (3280 x 2464 et 4608 x 2592)
 - Raspberry Pi 0 : Pas assez de mémoire ✗
 - Raspberry Pi 5 : Temps de rafraîchissement trop élevé ✗
 - Bonne précision ✓
- Résolutions moyennes : (1640 x 1232 et 1536 x 864)
 - Raspberry Pi 0 : Temps de rafraîchissement trop élevé ✗
 - Raspberry Pi 5 : Temps de rafraîchissement idéal ✓
 - Bonne précision ✓

Choix d'une caméra

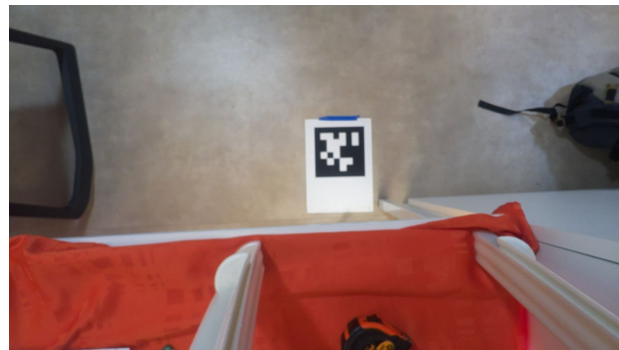
- Pi Camera V3 Wide : Plus large champ de vision malgré le recadrage ✓
- Pi Camera V2 : Précision légèrement meilleure ($\approx 5\text{mm}$), mais champ de vision légèrement moins bon 🟡
- Pi Camera V3 : Faible champ de vision ✗



Pi Camera V3
(recadrage)



Pi Camera V2



Pi Camera V3 Wide
(recadrage)

Masse et consommation, choix d'une carte

Masse et consommation		
Carte	Masse (g)	Consommation moyenne (W)
Pi 0	12	1.8
Pi 5	66	7

Problème : Masse de la Pi 5 trop élevée, et puissance de calcul de la Pi 0 trop faible.

Solution :

- **Compute module**, puissance d'une Pi 5 mais masse de **16g** !
- Inconvénient : Nécessite un PCB supplémentaire.

Bilan

- Configuration choisie :
 - Carte : Pi 5 Compute Module
 - Caméra : Pi Camera V3 Wide
 - Résolution : 1536 x 864 px

Performances attendues	
Erreur absolue de position à 1,275 m du sol	0-1 cm selon x et y, et 0-8 cm selon z
Erreur absolue d'angle à 1,275 m du sol	0-3° sur les 3 angles
Temps de rafraîchissement	0,05 s, soit 20 Hz
Consommation	7 W
Masse	20 g

Pistes d'améliorations

- Calibration
- Position de la caméra
- Taille des tag
- Nombre de tags (moyennage des positions)

Bibliographie

Localisation via la vision avec la bibliothèque Apriltags, Gilles Arthur FADE, Anthony JUTON, Revue 3EI

<https://april.eecs.umich.edu/software/apriltag>

https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html

https://github.com/elenagiraldo3/april_tags_autolocalization

https://docs.opencv.org/4.x/d5/d1f/calib3d_solvePnP.html

Comparaison des différentes solutions

Solution	Avantages	Inconvénients	Précision
Accéléromètre	Léger, compact, facile à utiliser	Mesure uniquement une variation de position, risque de dérive, ne peut être utilisé seul	$\pm 2^\circ$ pour le lacet
GPS-Indoor (ultrason)	Précis, insensible aux perturbations électromagnétiques	Obstruction possible des ondes sonores, nécessite d'installer des balises actives, coût, masse	± 2 cm
Télémètre (infrarouge)	Rapide, petit, léger	Portée limitée à 8 m, sensible à la nature du sol, mesure sur 1 axe uniquement	± 1 cm
Vision (Apriltags)	Position complète avec orientation, tags peu chers et faciles à installer	Obstruction possible des tags, sensible à luminosité, nécessite l'installation de tags	± 1 cm selon x et y, ± 10 cm selon z $\pm 3^\circ$ pour les 3 angles