

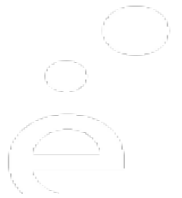
# Rappels



Module 1 : Présentation de la robotique mobile et des capteurs utilisés

Module 2 : Localisation par intégration odométrique

**Module 3 : Localisation par télémétrie laser**



# Classification des problèmes de localisation

- Dans toute *localisation probabiliste* basée carte, la « **croyance** » sur l'état du robot (*belief state*, en anglais) est représentée par une *pdf*

## 1 – Problème de localisation incrémentale

- On assume que la pose initiale  $p(0)$  du robot soit connue et que l'incertitude associée soit *petite*
- La pose courante du robot ( $p(t)$ ) est mise à jour en utilisant la connaissance de la pose précédente (“suivi ou tracking de pose”)
- La croyance sur l'état du robot est typiquement modelée avec une *pdf unimodale*, comme par ex. une pdf gaussienne



# Classification des problèmes de localisation

## 2 – Problème de localisation globale

- On assume que la pose initiale du robot ne soit pas connue. Ça veut dire que le robot peut être placé partout dans l'environnement et il doit se localiser
- La croyance initiale sur l'état du robot est typiquement une *pdf uniforme*

~~$p(0)$~~

$p(t) ?$

## 3 – Problème du robot kidnappé

- Le robot est kidnappé et déplacé en un autre endroit
- Ce problème est similaire au problème de localisation globale seulement si le robot se rend compte d'être été kidnappé

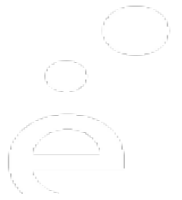


# Classification des problèmes de localisation

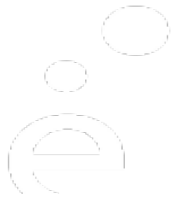


## Localisation de Markov (resolution des problèmes 1, 2 et 3)

- La croyance sur l'état du robot peut être représentée par une *pdf arbitraire*
- Localisation à partir de *toute position initiale inconnue*
- *Approche récursive*: phase de *prédiction* et de *correction*
- On peut sortir de *situations ambiguës* (en effet, plusieurs positions possibles du robot sont “suivies” en même temps)
- Pour mettre à jour la probabilité de toutes les positions dans tout l'espace des états à chaque instant, on a besoin d'une *représentation discrète* de l'espace (par ex. une grille avec cellules de taille fixe ou un graphe topologique)
- Puissance de calcul et mémoire requise *importantes* (par conséquent, la résolution et la taille de la carte sont limitées)



# Classification des problèmes de localisation



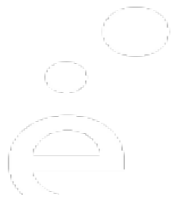
## Localisation par filtre de Kalman (resolution du problème 1)

- La position initiale du robot est *connue*
- La croyance sur l'état du robot est représentée par une seule *pdf gaussienne*. Il donc suffit de mettre à jour à chaque itération les paramètres de la gaussienne (c'est-à-dire, la *moyenne* et la *matrice de covariance*)
- Localisation *precise* and *efficace*
- Utilisable avec une *représentation continue* de l'environnement
- Si l'incertitude sur la position du robot devient trop importante (par ex. suite à la collision avec un obstacle) et la pdf n'est *plus vraiment unimodale*, le filtre de Kalman peut n'être pas capable de capturer la multitude de positions possibles du robot et il peut échouer (par conséquent, le robot sera perdu à jamais)

# Classification des problèmes de localisation



## Localisation incrémentale



- *Hypothèse*: position initiale connue précisément
- Le robot mobile se déplace: l'odométrie permet de *mesurer* son mouvement
  - *Perception proprioceptive*
  - Croissance permanente de l'incertitude de localisation
  - Pour borner cette incertitude:
    1. Localisation par rapport à une *carte* (« *M* »)
    2. Observation de l'environnement (*perception extéroceptive*)
      - Laser, ultrasons, caméras, etc.

# Autres techniques de localisation

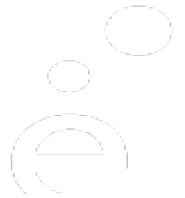


## Localisation Monte Carlo

- Filtre de Kalman
  - Pdf gaussienne
- Filtre particulaire (*particle filter\**, en anglais)
  - Pas limité à pdf unimodales

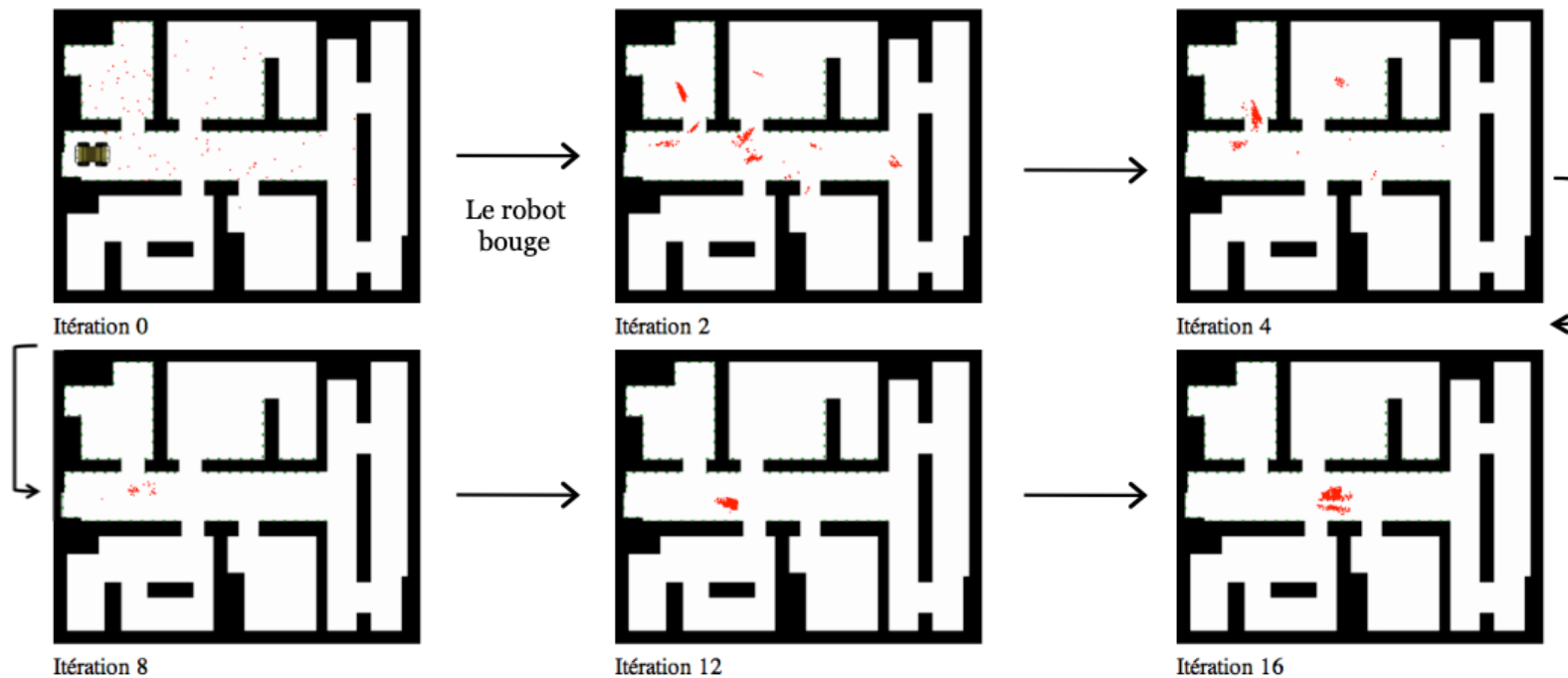
**Localisation Monte Carlo:** le filtre particulaire est utilisé pour représenter la croyance sur l'état du robot

- Ensemble de  $N$  poses initiales dans une carte
- On fait évoluer ces  $N$  poses possibles en fonction des perceptions proprioceptives et extéroceptives via un modèle probabiliste
- Quand les  $N$  poses (à savoir, les  $N$  particules), convergent vers un même état, la localisation est la plus sûre possible



# Autres techniques de localisation

État initiale:  $N$  particules (rouges)





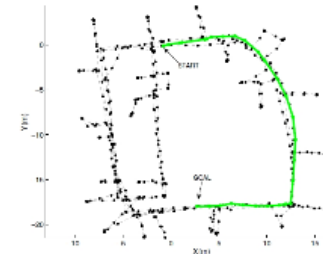
# Autres techniques de localisation



## Construction autonome de carte

- SLAM (*Simultaneous Localization And Mapping*)
  - Localisation et cartographie simultanées (pas de connaissance *a priori* de carte)
  - Solution efficace du SLAM: «Saint Graal» de la robotique mobile
  - Formalisme du *filtre de Kalman étendu* (« EKF SLAM »)
    - On estime:
      - L'état du robot et son incertitude
      - L'état et l'incertitude de chaque primitive (point, droite, etc.) perçue de l'environnement

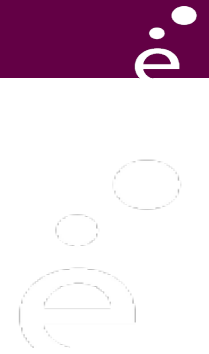
**Recherche très active:** SLAM visuel (une caméra: MonoSLAM, LSD-SLAM, ORB-SLAM, OpenVSLAM, PTAM), SLAM basé filtre particulaire, SLAM coopératif ou C-SLAM, « Pose Graph Optimization », etc.



# Autres techniques de localisation

## Localisation basée balises

- Balise/amer (*beacon* ou *landmark*, en anglais)
  - Objet *passif* ou *actif* dans le champ de perception du robot
  - La position globale des balises est *connue* avec précision
- a. Balise passive
  - Considérée avec la localisation par filtre de Kalman
  - *Problème*: répartition judicieuse des balises dans l'environnement



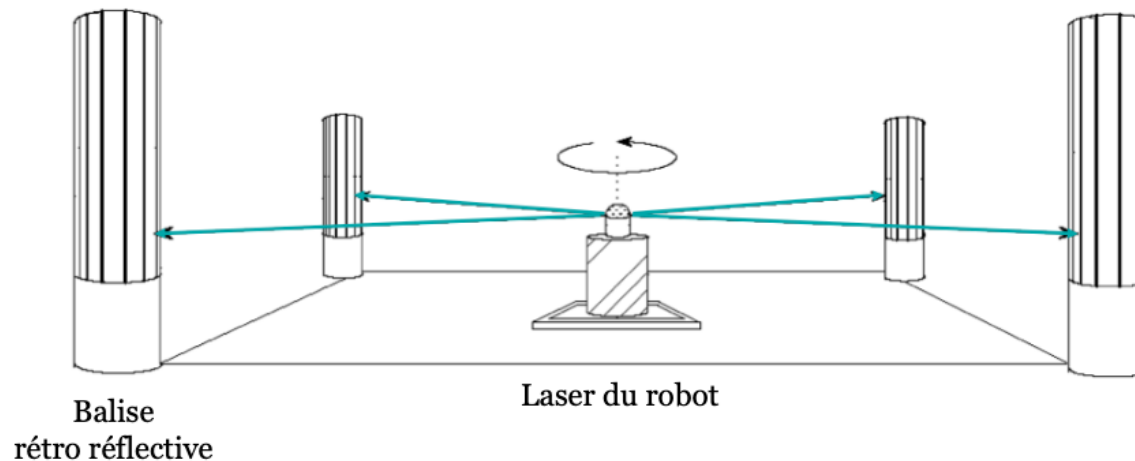
# Autres techniques de localisation



## Localisation basée balises

### *Exemple 1 : balises rétro réfléchives*

- Robot équipé avec une nappe laser
- Mesure d'énergie renvoyée vers le robot
- 3 balises minimum (moins si l'odométrie est disponible)



# Autres techniques de localisation



## Localisation basée balises

### *Exemple 2 : balises colorées*

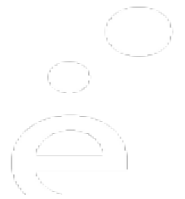
- Chaque balise a un code couleur unique
- Le robot est équipé avec une caméra (standard ou panoramique) qui permet de détecter les balises



Robot avec caméra embarquée



Balises avec code couleur  
(3 couleurs: noir, jaune, rose)



# Autres techniques de localisation



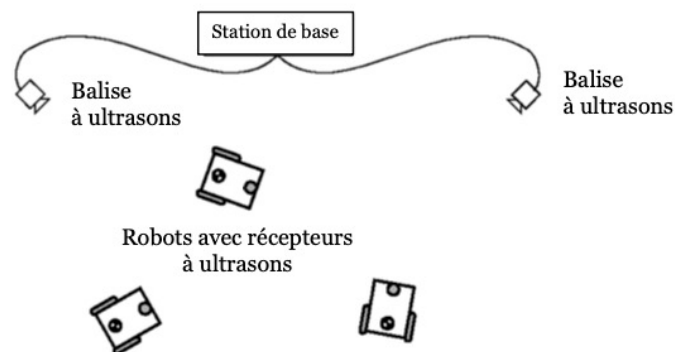
## Localisation basée balises

### b. Balise active

- Applications industrielles et militaires
- Localisation robuste ...  
... mais pré-équipement coûteux de l'environnement

#### *Exemple : balises à ultrasons*

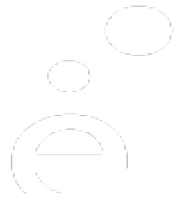
- Localisation précise
- Peu flexible !



# Autres techniques de localisation



## Localisation basée chemin



- Encore plus fiable que les balises
- Le chemin du robot est explicitement tracé
  - Localisation relative au chemin

### *Exemples:*

- Tracé de peinture UV réfléchive
- Câble guide sous le sol détecté par induction électromag.

### **Inconvénients**

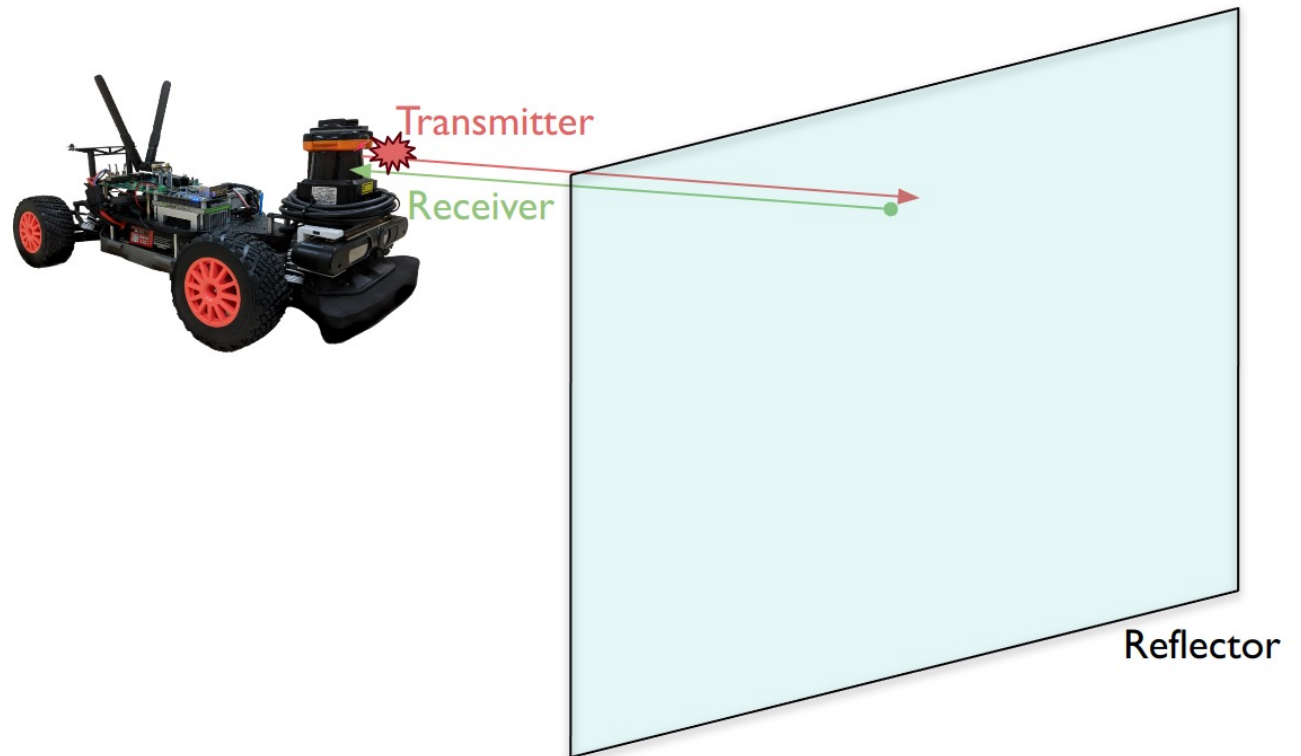
- Encore moins flexible que les balises actives
- Le robot ne doit pas trop dévier du chemin prévu



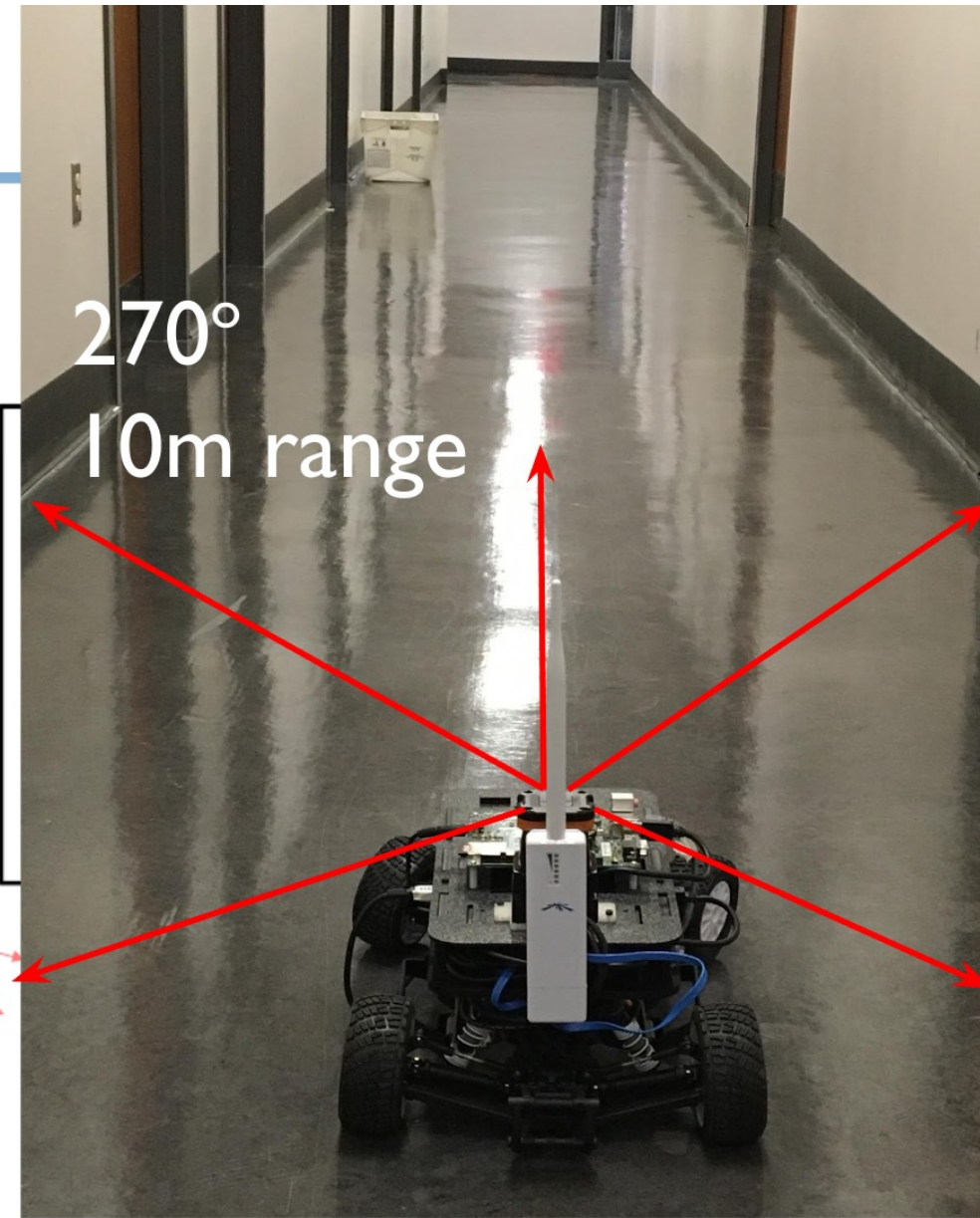
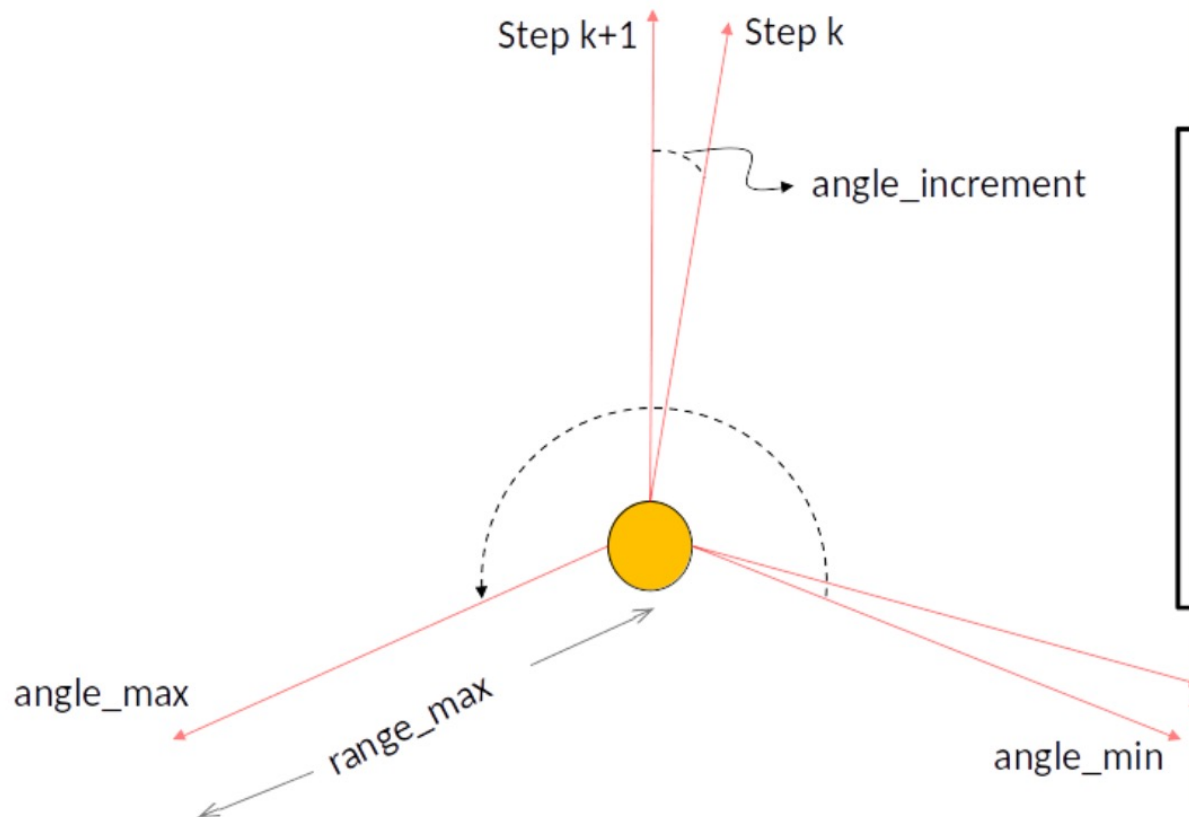
# Planar LIDAR

Distance to obstacle = (speed of light \* time traveled)/2

- Maximum range is constrained by laser energy
  - Need a large Signal-to-Noise ratio
- Energy of reflected ray depends upon:
  - Divergence of laser
  - Bouncing reflections
  - Humidity
  - Reflectivity of target
  - Detector sensitivity



# Planar LIDAR

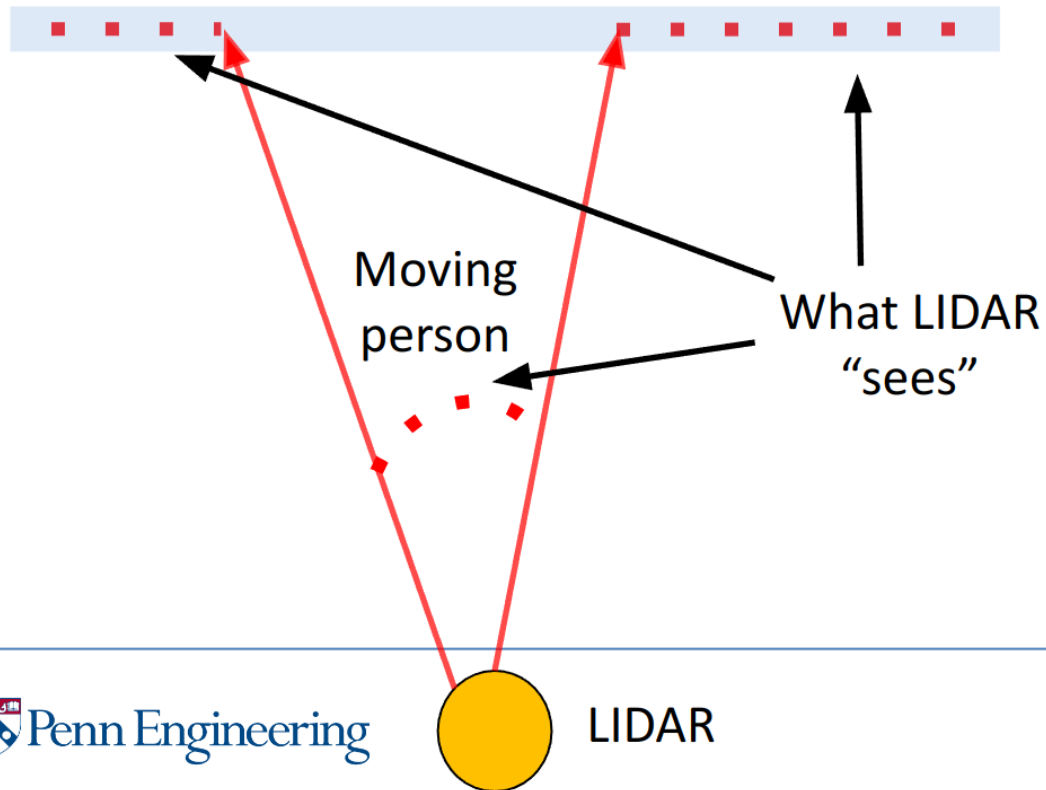




# A simple scan sequence

## TOP VIEW

Whiteboard

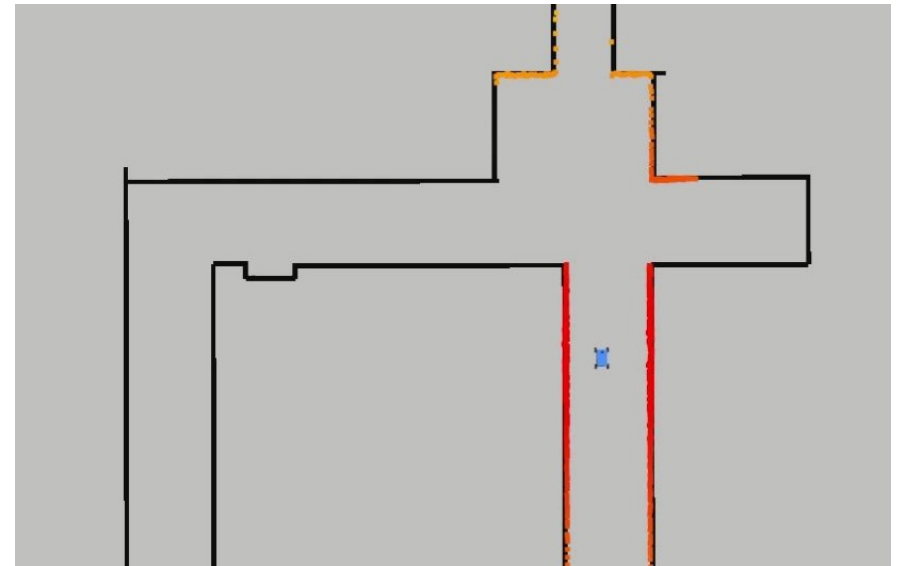


# Reprojection des données



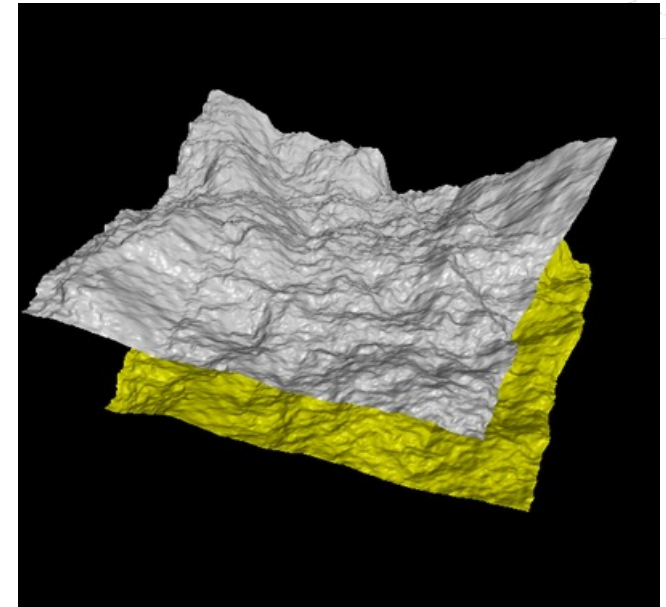
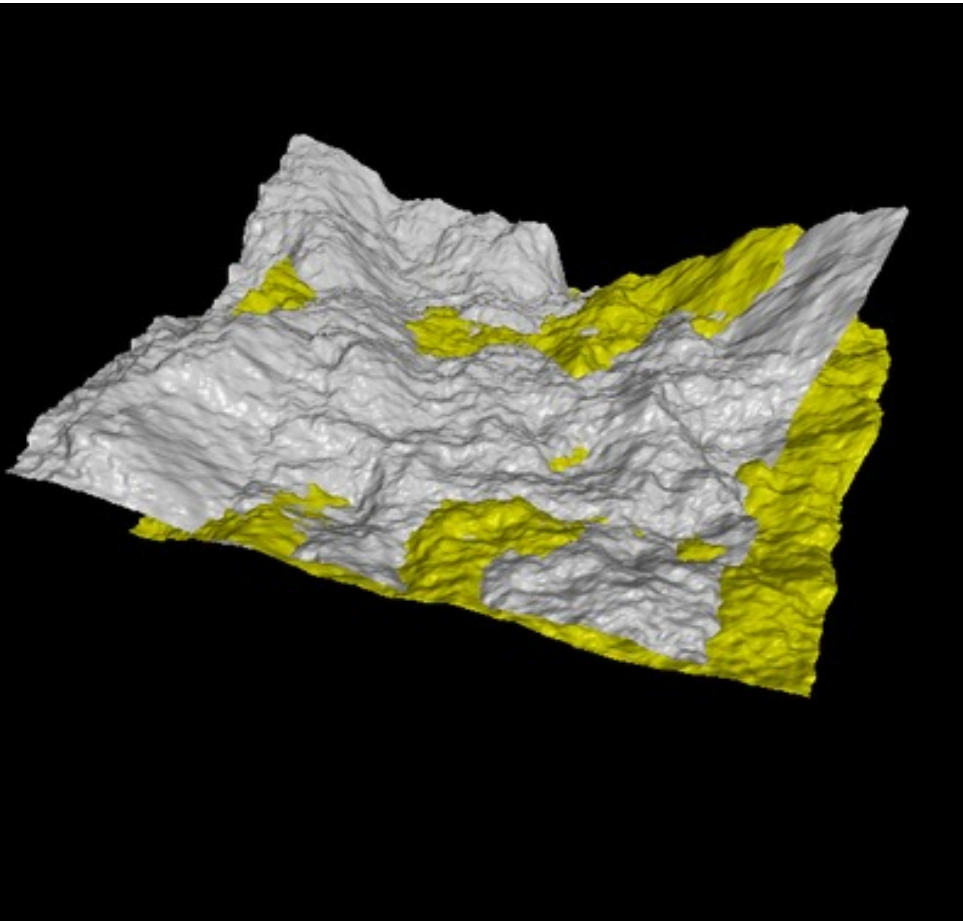
Les données du télémètre laser sont exprimées dans le repère robot.

Il est nécessaire de réaliser un changement de repère : Robot  $\rightarrow$  Global pour pouvoir les afficher par superposition avec la carte de l'environnement.



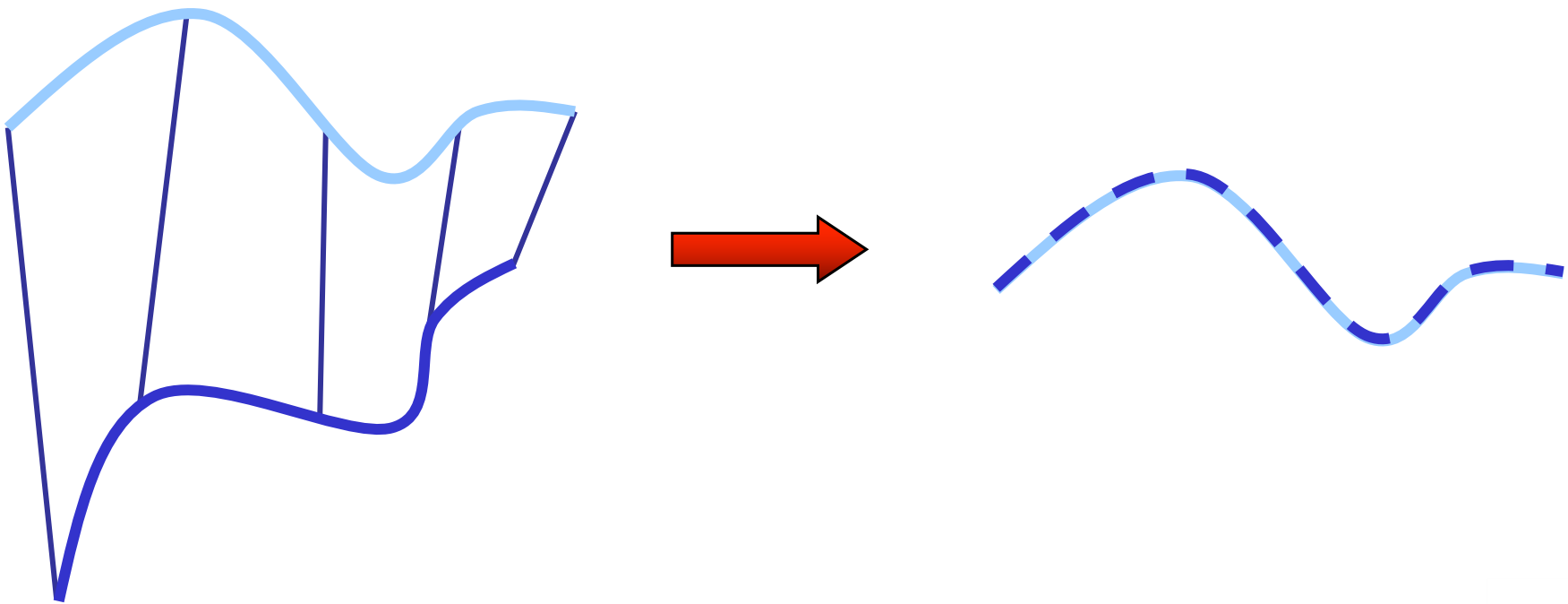
# Recalage de nuage de points par Iterated Closest Points (ICP)

e.



# Recalage de nuage de points par Iterated Closest Points (ICP)

Si les correspondances correctes sont connues, la rotation/translation relative correcte peut être calculée.



# Recalage de nuage de points par Iterated Closest Points (ICP)

- Given: two corresponding point sets:

$$X = \{x_1, \dots, x_n\}$$

$$P = \{p_1, \dots, p_n\}$$

- Wanted: translation  $t$  and rotation  $R$  that minimizes the sum of the squared error:

$$E(R, t) = \frac{1}{N_p} \sum_{i=1}^{N_p} ||x_i - Rp_i - t||^2$$

Where  $x_i$   $p_i$  are corresponding points.

# Recalage de nuage de points par Iterated Closest Points (ICP)



$$\mu_x = \frac{1}{N_x} \sum_{i=1}^{N_x} x_i \quad \text{and} \quad \mu_p = \frac{1}{N_p} \sum_{i=1}^{N_p} p_i$$

are the centers of mass of the two point sets.

## Idea:

- Subtract the corresponding center of mass from every point in the two point sets before calculating the transformation.

$$X' = \{x_i - \mu_x\} = \{x'_i\}$$

- The resulting point sets are:

$$P' = \{p_i - \mu_p\} = \{p'_i\}$$

# Recalage de nuage de points par Iterated Closest Points (ICP)

Let  $W = \sum_{i=1}^{N_p} x_i' p_i'^T$

denote the singular value decomposition (SVD) of  $W$  by:

$$W = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} V^T$$

where  $U, V \in \mathbb{R}^{3 \times 3}$  are unitary, and

$\sigma_1 \geq \sigma_2 \geq \sigma_3$  are the singular values of  $W$ .

# Recalage de nuage de points par Iterated Closest Points (ICP)

If  $\text{rank}(W) = 3$ , the optimal solution of  $E(R,t)$  is unique and is given by:

$$R = UV^T$$
$$t = \mu_x - R\mu_p$$

The minimal value of error function at  $(R,t)$  is:

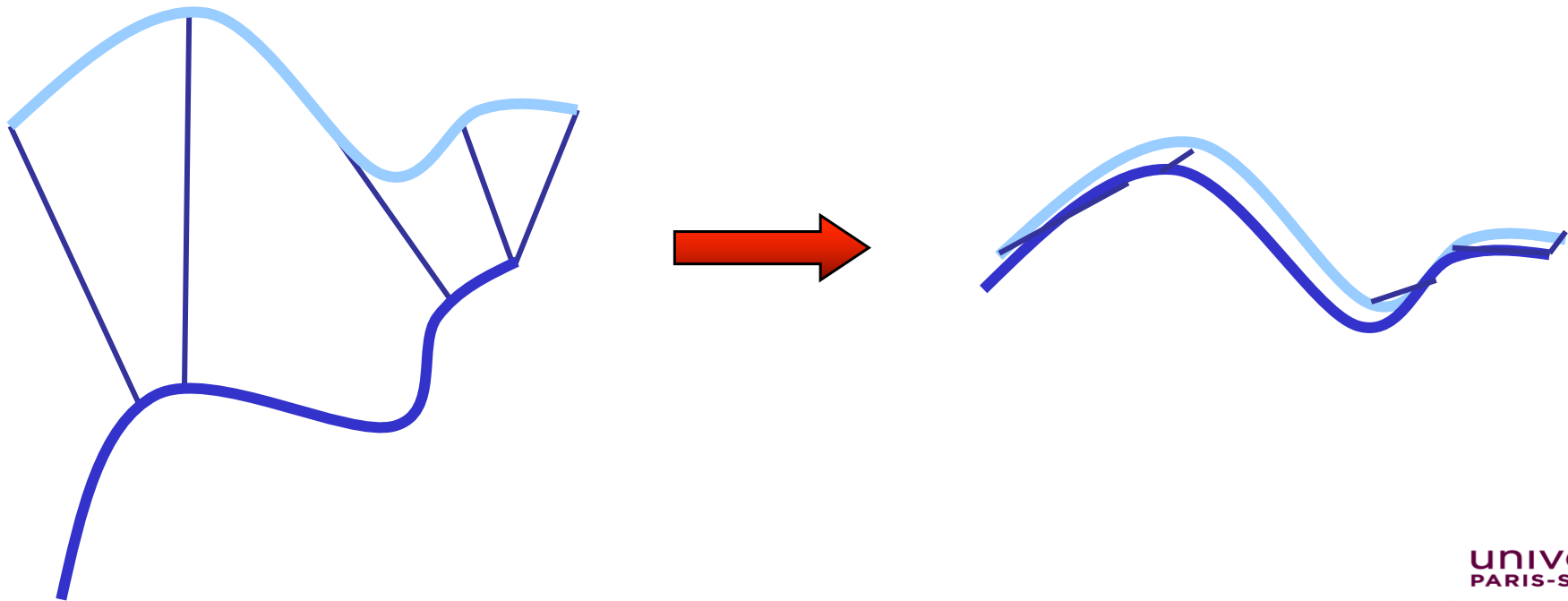
$$E(R, t) = \sum_{i=1}^{N_p} (||x'_i||^2 + ||y'_i||^2) - 2(\sigma_1 + \sigma_2 + \sigma_3)$$



# Recalage de nuage de points par Iterated Closest Points (ICP)

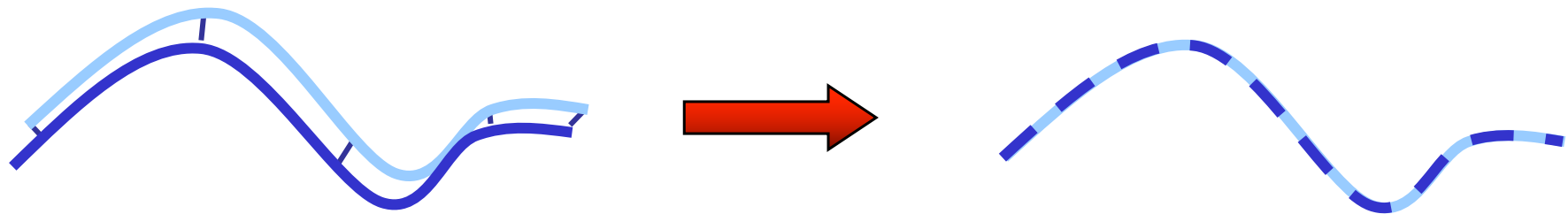


- If correct correspondences are not known, it is generally impossible to determine the optimal relative rotation/translation in one step



# Recalage de nuage de points par Iterated Closest Points (ICP)

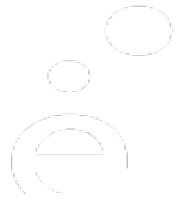
- Idea: iterate to find alignment
- Iterated Closest Points (ICP)  
[Besl & McKay 92]
- Converges if starting positions are  
“close enough”



# Webots – Simulateur (Thymio)



Sujet disponible sur Ecampus



- Prendre en main le télémètre laser embarqués sur le Thymio
- Réaliser des changements de repère pour re-projeter les données
- Programmer une localisation par ICP