

Navigation pour les systèmes autonomes

Cours ENSTA Paris - ROB312 / 05

David FILLIAT

david.filliat@ensta-paris.fr

2023-2024



Problématique de la cartographie

Cartographie incrémentale

- Cartographie topologique
- Corrélation
- Grilles d'occupation

Cartographie avec retour arrière

- EKF SLAM

Problématique de la cartographie



Cartographie

recherche de la carte *la plus probable*, connaissant les données perçues par le robot, dans l'espace des cartes de la représentation choisie

→ Espace beaucoup plus grand que l'espace utilisé pour la localisation

Localisation associée à la cartographie

→ recherche dans un espace ouvert

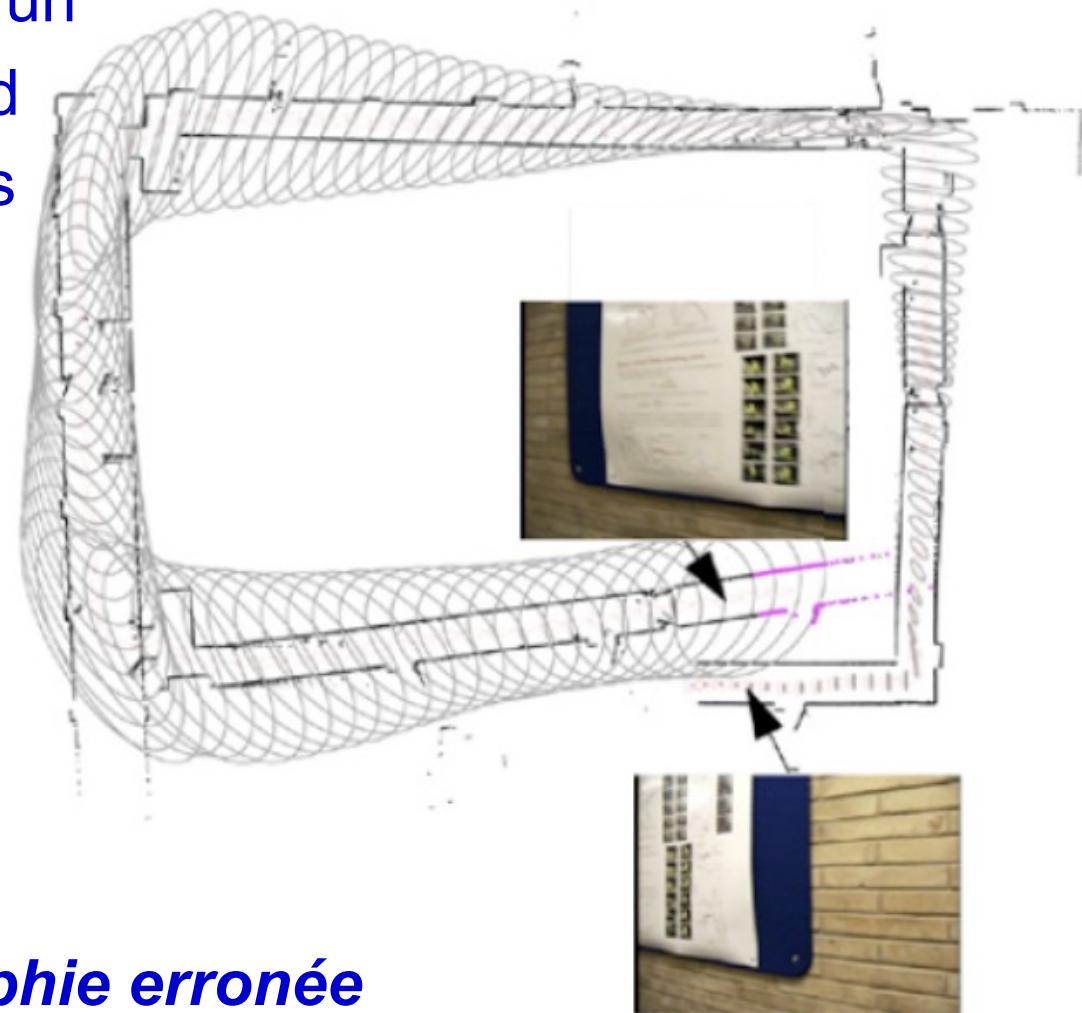
Carte incomplète

→ utilisation fréquente de suivi de position (pas de correction lors de la découverte d'environnement inconnu)
→ limitations, notamment pour les grands cycles

Environnement contenant un cycle beaucoup plus grand que la portée des capteurs

Dérive des méthodes de localisation (suivi de position)

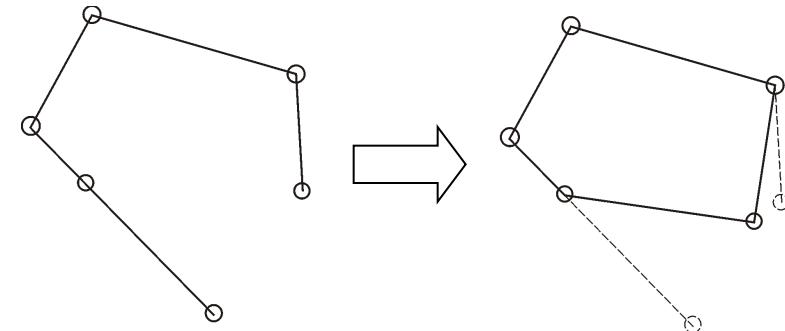
→ **cartographie erronée**



P. Newman, Oxford Mobile Robotics Group

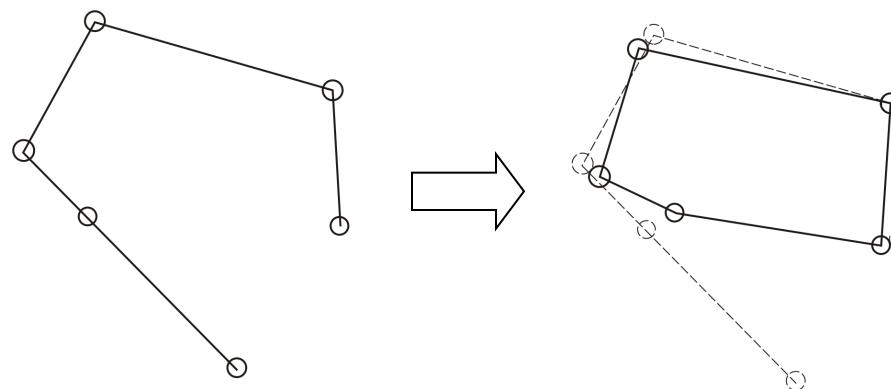
Cartographie incrémentale

- Pas de reprise des informations passées si elle se révèlent erronées
→ modifications locales de la carte

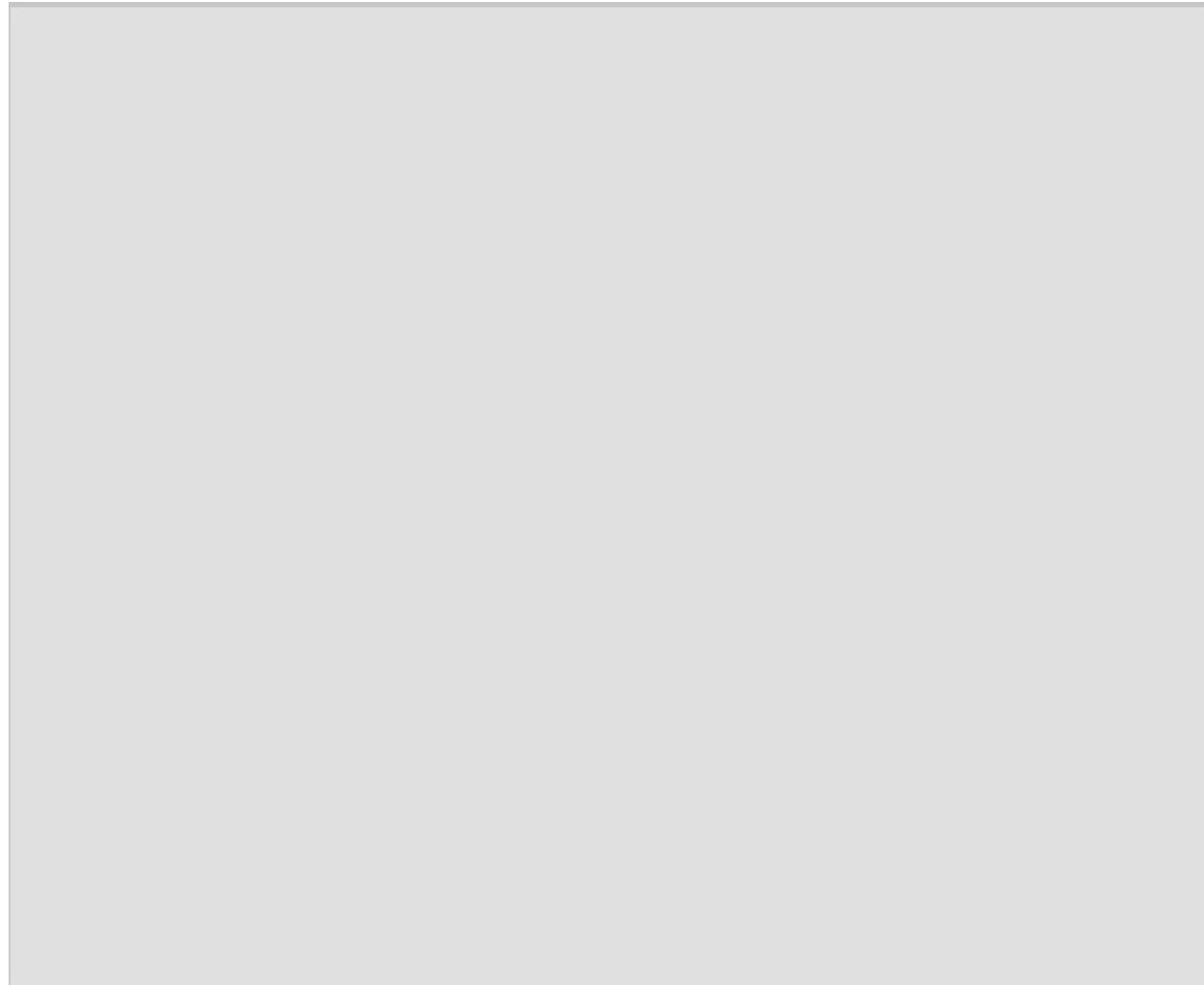


Cartographie avec retour en arrière

- Lorsque la localisation se révèle fausse, il existe une méthode pour reprendre les modifications déjà faites
→ possibilité de modifications globales



Raw Data



Robotics and State Estimation Lab (Dieter Fox)

<http://rse-lab.cs.washington.edu/>

Cartographie avec retour arrière



Robotics and State Estimation Lab (Dieter Fox)

<http://rse-lab.cs.washington.edu/>

Cartographie incrémentale



ENSTA



Cartographie topologique

- Algorithme simple, dépend beaucoup de la qualité de la localisation (essentiellement par les perceptions)

Algorithm 4 Algorithme de cartographie topologique pour un déplacement u et des perceptions Y

- 1: **if** Il existe $noeud_i$ compatible avec u , Y et $Position_{t-1}$ **then**
- 2: $Position_t = noeud_i$
- 3: **else**
- 4: $Position_t = \text{nouveau noeud}$
- 5: **end if**
- 6: Mettre à jour données de $Position_t$ avec Y
- 7: Mettre à jour la connexion $Position_{t-1} - Position_t$ avec u

Corrélation de scans (pas ex. ICP)

- Utilisable en environnement inconnu
- Hyp : large recouvrement entre scans successifs

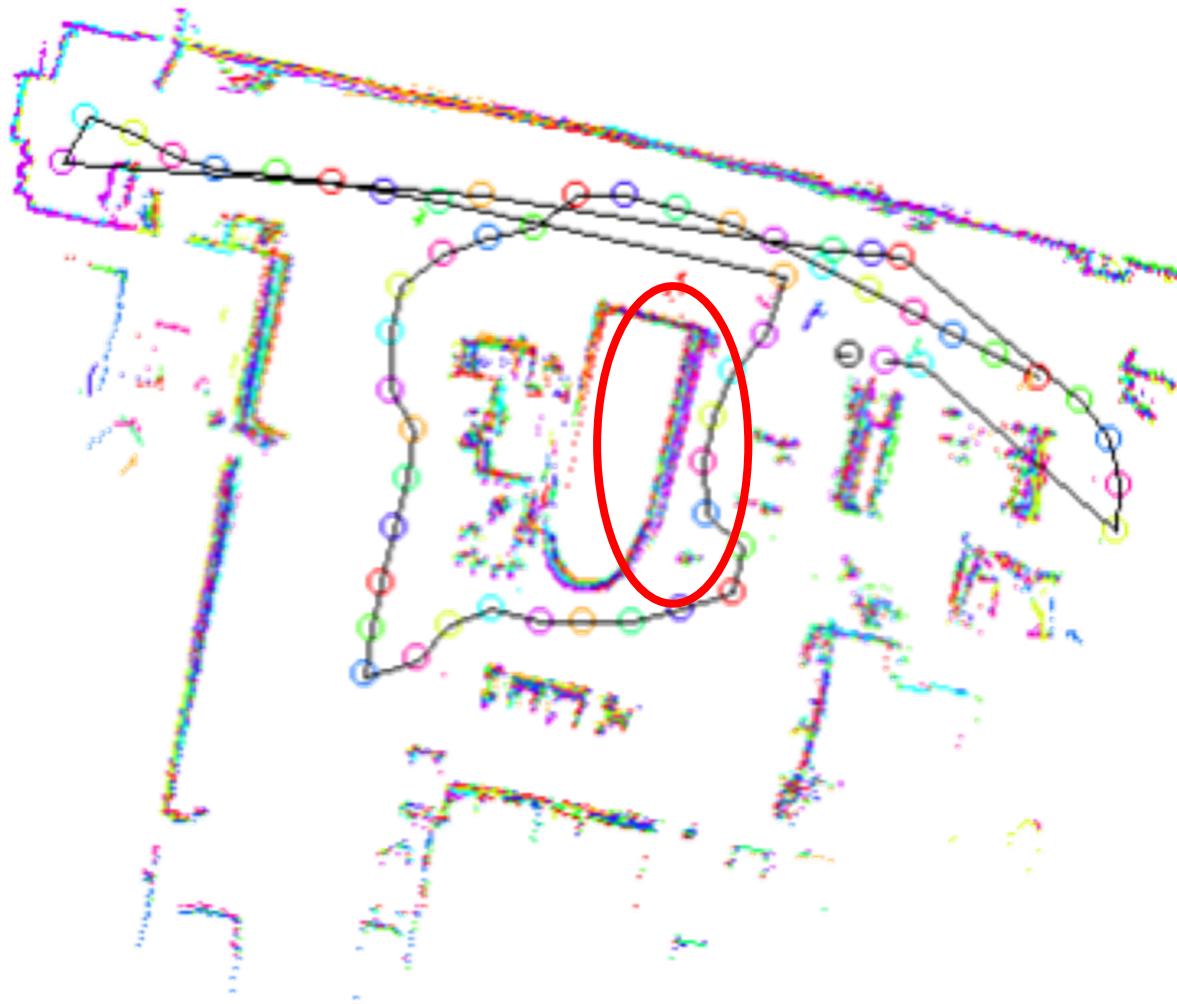
Algorithme

- Corrélation avec le scan précédent -> localisation
- Corrélation avec le scan de la carte le plus proche
 - > recalage lors de fermetures de boucles
- Ajout du scan à la carte si on est assez loin de tous les autres



Limitation pour les grands cycles

- Gestion des fermetures de boucles (CF plus loin)



Probabilité de présence d'obstacles

- Représentée sur grille régulière

Position supposée connue

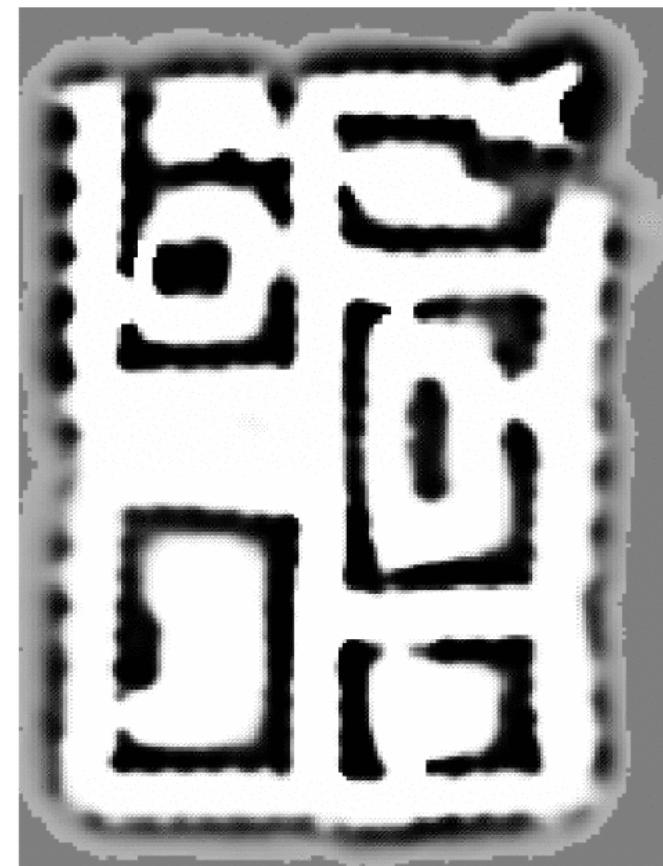
ou *Position estimée par suivi de position*

- Corrélation de scans
- Corrélation de grille locale avec la grille globale
- Hypothèses sur l'environnement (murs //)

On estime : $P(\text{occ}_i | s_1, \dots, s_T)$

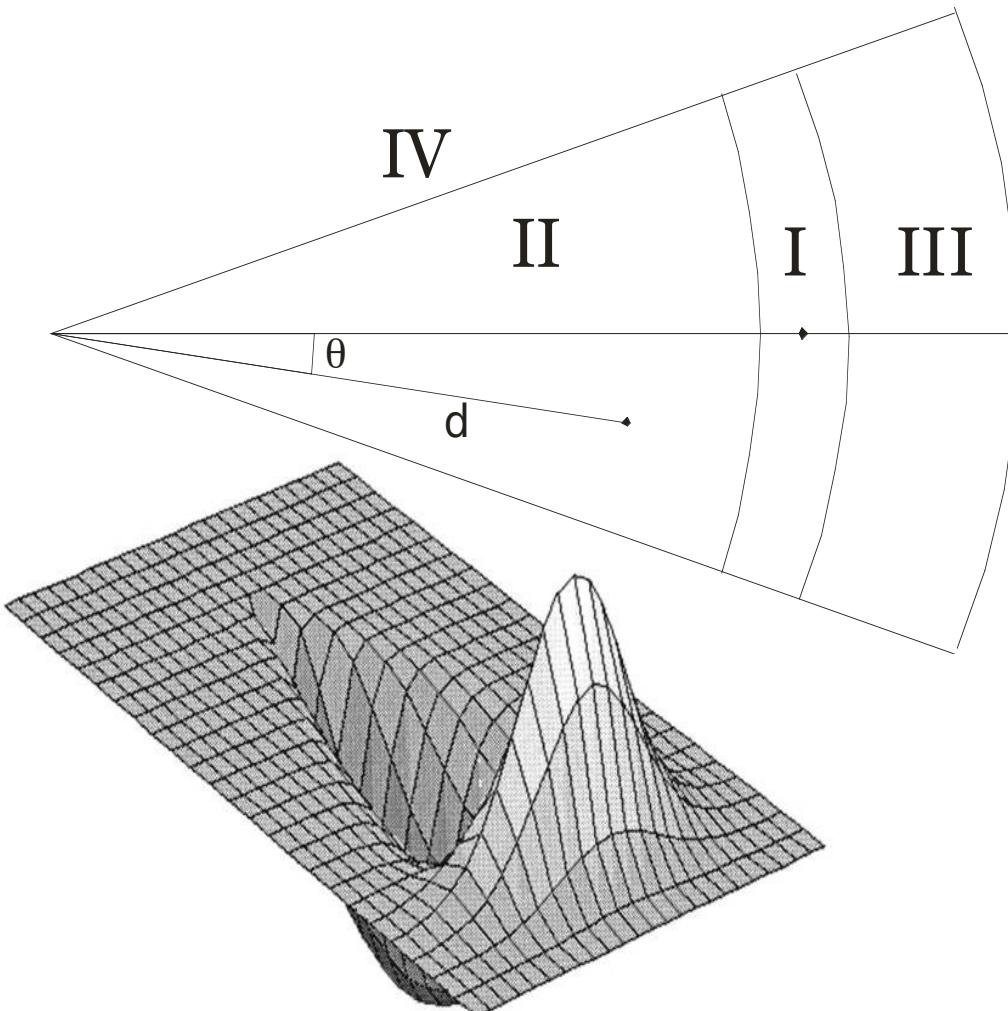
en connaissant : $P(\text{occ}_i | s)$ (modèle inverse)

Position connue !



Rappel

- $P(\text{valeur réelle}|\text{mesure}) :$

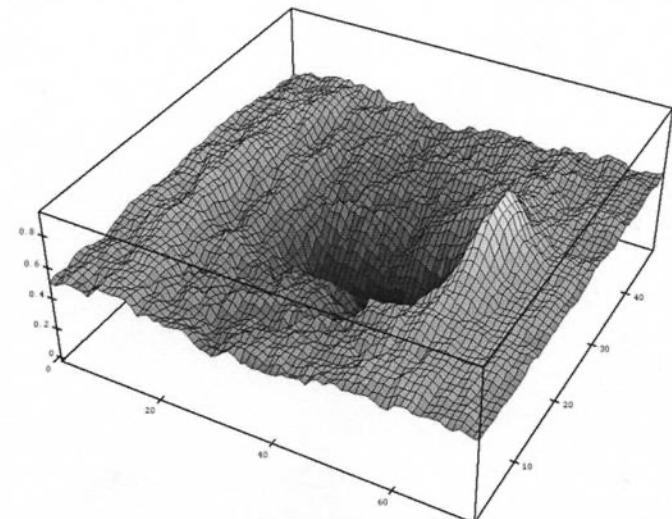


$$\text{I : } \text{Gauss}(\theta) * \text{Gauss}(d-d_m)$$

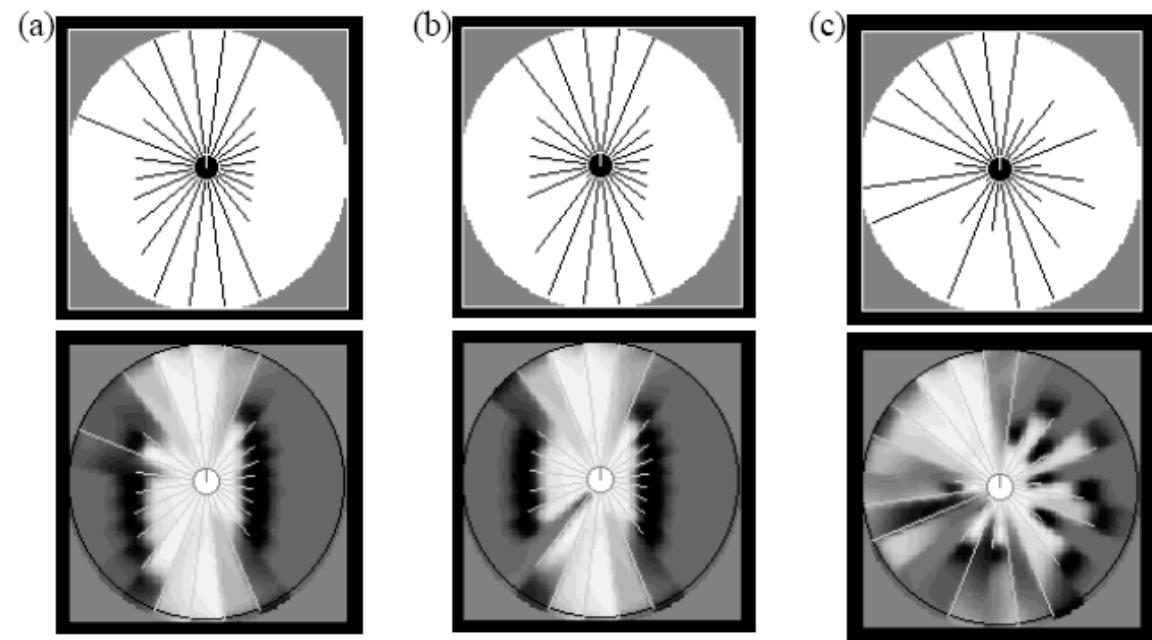
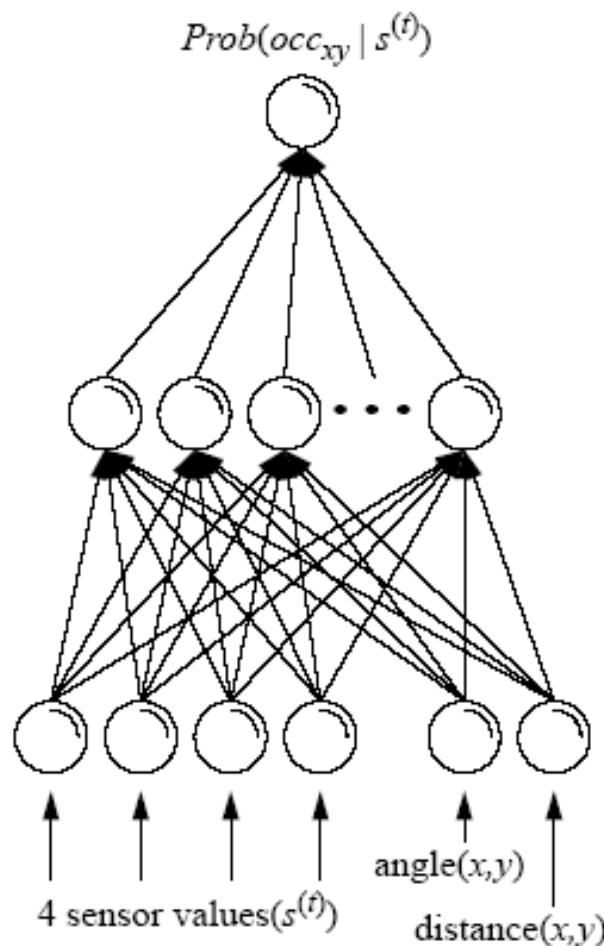
$$\text{II : } \exp(d) * \exp(-\theta)$$

$$\text{III : } \exp(d) * \exp(-\theta)$$

$$\text{IV : } 0$$



Variante : Réseau de neurone utilisant plusieurs mesures (Thrun 98) :



Mise à jour de la probabilité d'une cellule dans le champ du capteur :

- Utilisation d'une valeur intermédiaire:

$$l_i^T = \log \left(\frac{P(\text{occ}_i^T)}{1 - P(\text{occ}_i^T)} \right)$$

Le log permet de stocker des probabilités faibles avec moins d'approximations

- Mise à jour :

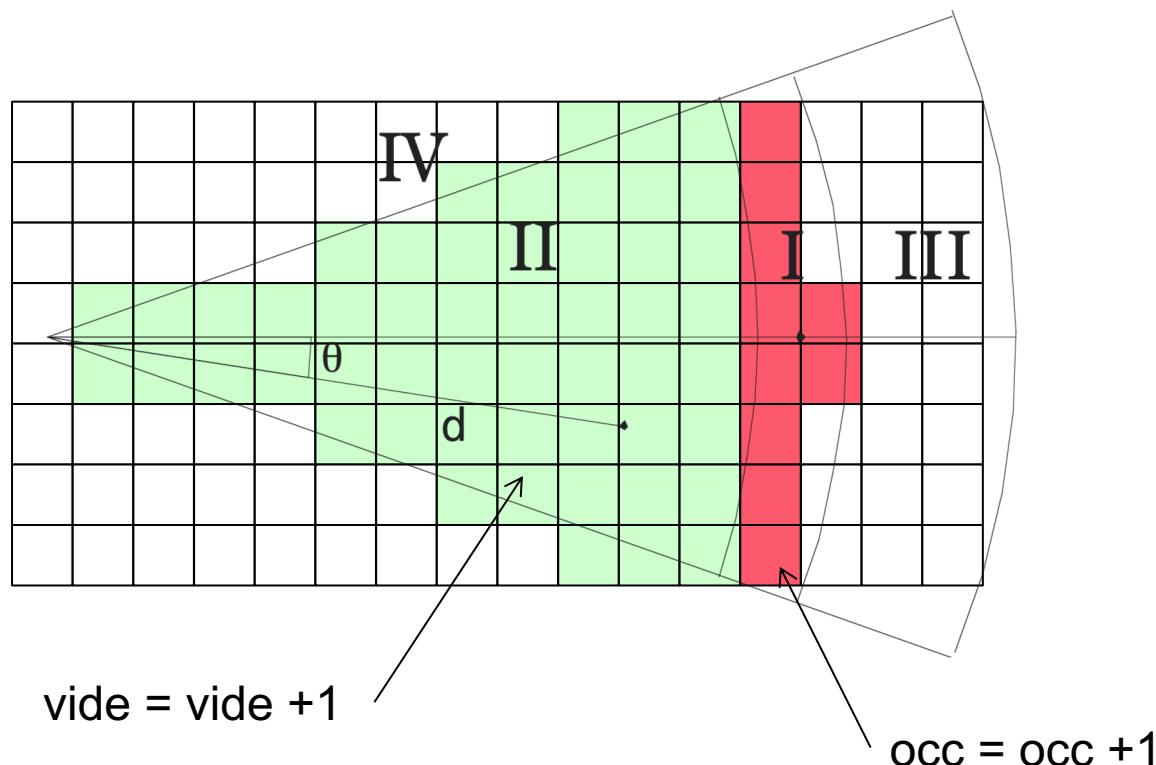
$$l_i^T = \log \left(\frac{P(\text{occ}_i|s_T)}{1 - P(\text{occ}_i|s_T)} \right) + l_i^{T-1}$$

- éventuellement :

$$p(\text{occ}_i^T) = 1 - \frac{1}{e^{l_i^T}}$$

Variante : simple décompte

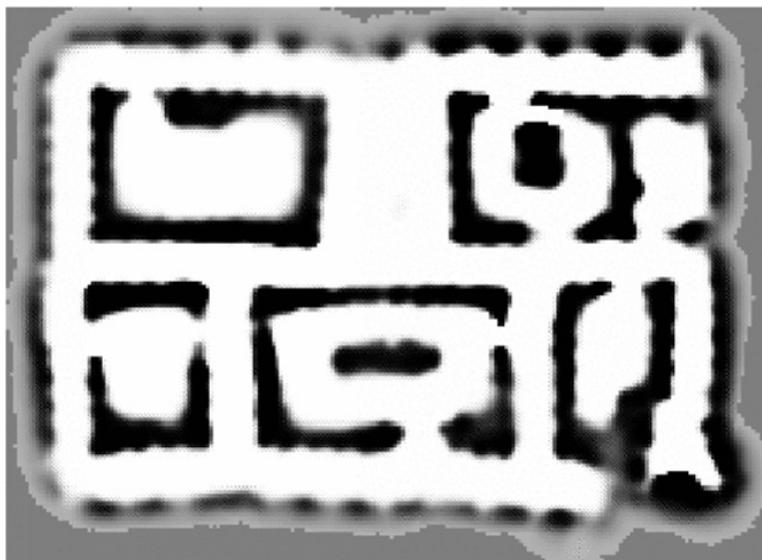
$$P_{occ}(x,y) = \frac{occ(x,y)}{vide(x,y) + occ(x,y)}$$



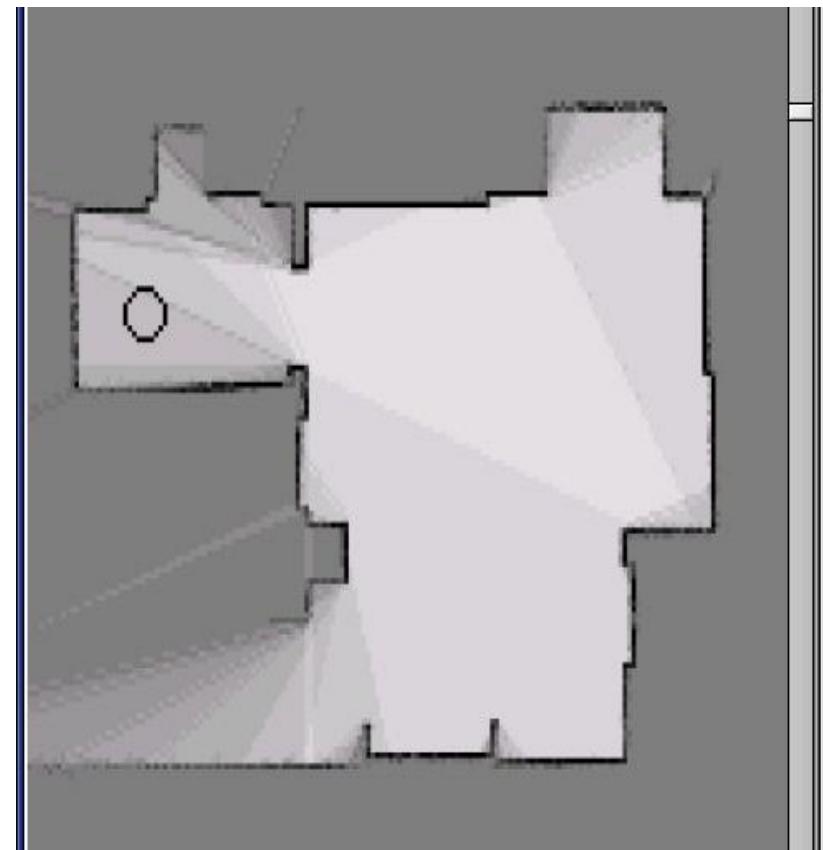
Exemples :

Utilisation de sonars
+ hypothèse murs orthogonaux

(Thrun 98)



Utilisation d'un télémètre laser
+ localisation par corrélation de scans



Autre exemple : HECTOR SLAM

Utilisation d'un télémètre laser

Localisation par rapport à la carte par descente de gradient

- Trouver position tq tt les pts du laser soient dans des cases occupées:

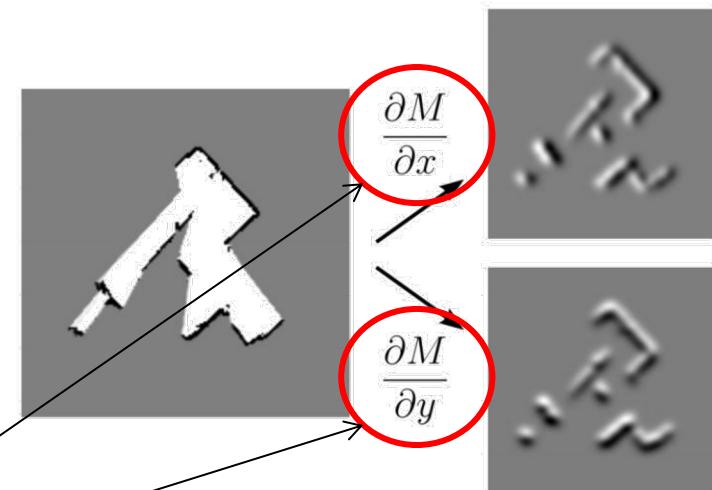
$$\xi^* = \underset{\xi}{\operatorname{argmin}} \sum_{i=1}^n [1 - M(\mathbf{S}_i(\xi))]^2$$

- Dvp de Taylor ordre 1 :

$$\sum_{i=1}^n [1 - M(\mathbf{S}_i(\xi + \Delta\xi))]^2 \rightarrow 0$$

- Méthode de Gauss-Newton

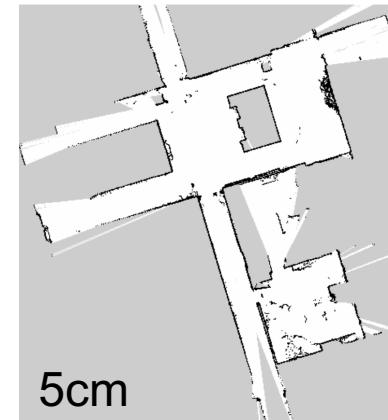
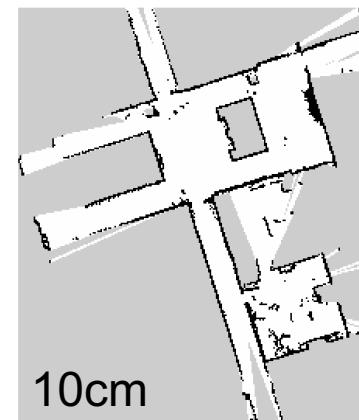
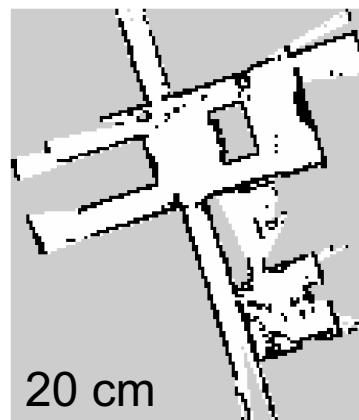
$$\Delta\xi = \mathbf{H}^{-1} \sum_{i=1}^n \left[\nabla M(\mathbf{S}_i(\xi)) \frac{\partial \mathbf{S}_i(\xi)}{\partial \xi} \right]^T [1 - M(\mathbf{S}_i(\xi))]$$



Autre exemple : HECTOR SLAM

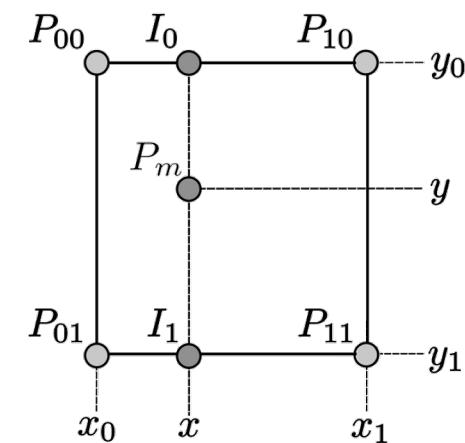
Quelques trucs supplémentaires:

Approche multi-échelle



Estimation continue via interpolation.

$$\begin{aligned} M(P_m) \approx & \frac{y - y_0}{y_1 - y_0} \left(\frac{x - x_0}{x_1 - x_0} M(P_{11}) + \frac{x_1 - x}{x_1 - x_0} M(P_{01}) \right) \\ & + \frac{y_1 - y}{y_1 - y_0} \left(\frac{x - x_0}{x_1 - x_0} M(P_{10}) + \frac{x_1 - x}{x_1 - x_0} M(P_{00}) \right) \end{aligned}$$



Autre exemple : HECTOR SLAM

S. Kohlbrecher and J. Meyer and O. von Stryk and U.Klingauf : A Flexible and Scalable SLAM System with Full 3D Motion Estimation.

ROS package `Hector_SLAM`



Limitations

- Perceptual aliasing -> topologie/structure incohérente
 - Nécessite une localisation métrique précise
 - Limité pour les environnements contenant de grands cycles
 - ⇒ fermetures non détectée
 - ⇒ erreurs lors de la fermeture des cycles
 - Impossibilité de retours en arrière sur des mises à jour déjà faites
- mais Modifications possibles de la carte selon nouvelles données

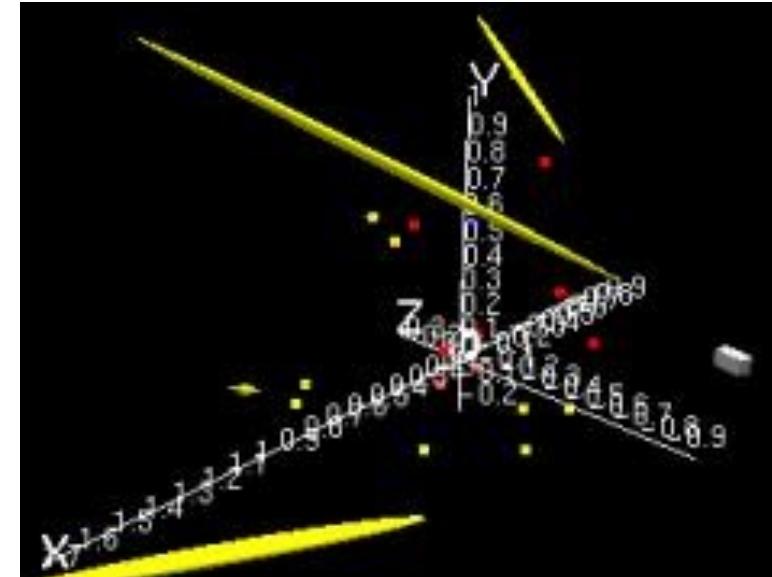
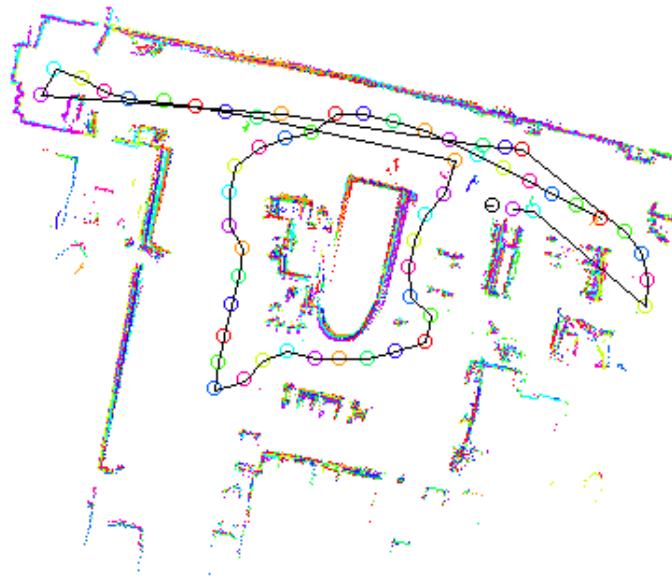
Cartographie avec retour en arrière

Méthodes de filtrage



Objectif :

- propager de nouvelles informations sur la position courante aux positions précédentes
- Mémorisation des relations positions du robot / éléments de la carte pour pouvoir les ré-utiliser
- Utilisation de carte d'amers ou d'une carte de scans



Cadre général

Filtrage Bayésien sur la position du robot et des éléments de la carte:

$$Bel(x_t, c_t) = \eta p(y_t | x_t, c_t) \int \int p(x_t, c_t | x_{t-1}, c_{t-1}, u_{t-1}) Bel(x_{t-1}, c_{t-1}) dx_{t-1} dc_{t-1}$$

Carte statique :

$$Bel(x_t, c_t) = \eta p(y_t | x_t, c_t) \int \int p(x_t | x_{t-1}, u_{t-1}) Bel(x_{t-1}, c_{t-1}) dx_{t-1} dc_{t-1}$$

Estimation complète de $Bel(x, c)$

→ approximations

- Filtrage de Kalman étendu
- Filtrage Particulaire

Simultaneous Localization and Mapping

- Filtre de Kalman étendu
- État = position du robot et des éléments de l'environnement (amers, scans..)

$$Bel(x_t, c_t) = \begin{bmatrix} x \\ y \\ \theta \\ x_{a1} \\ y_{a1} \\ \vdots \\ x_{aN} \\ y_{aN} \end{bmatrix}$$

Robot

Carte

Filtre de Kalman Etendu

$$X_{t+1} = f(X_t, u_t) + \varepsilon_{evo}$$

$$Y_t = h(X_t) + \varepsilon_{obs}$$

$$A_{ij} = \frac{\partial f_i}{\partial x_j}$$

$$B_{ij} = \frac{\partial f_i}{\partial u_j}$$

$$H_{ij} = \frac{\partial h_i}{\partial x_j}$$

$$X_t^* = f(X_{t-1}, u_t)$$

$$P_t^* = A.P_{t-1}.A^T + B.Q.B^T$$

$$Y_t^* = h(X_t^*)$$

$$K = P_t^* H^T . (H.P_t^*.H^T + P_Y)^{-1}$$

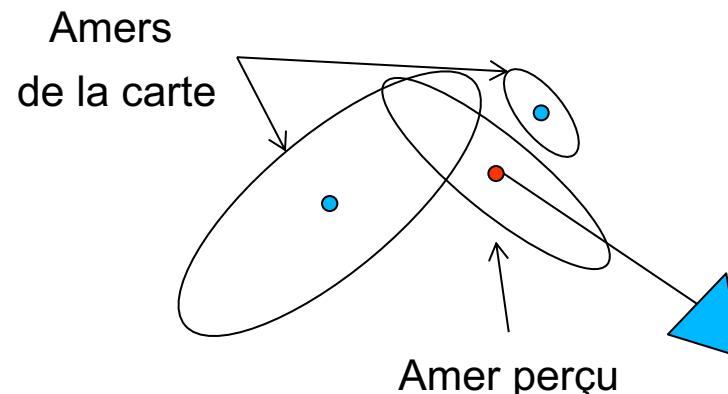
$$\hat{X}_t = X_t^* + K(Y_t - Y_t^*)$$

$$\hat{P}_t = P_t^* - KHP_t^*$$

- Les jacobiniennes se font aussi par rapport à la position des amers
- La mise à jour modifie à la fois la position du robot et des amers

Gestion des amers

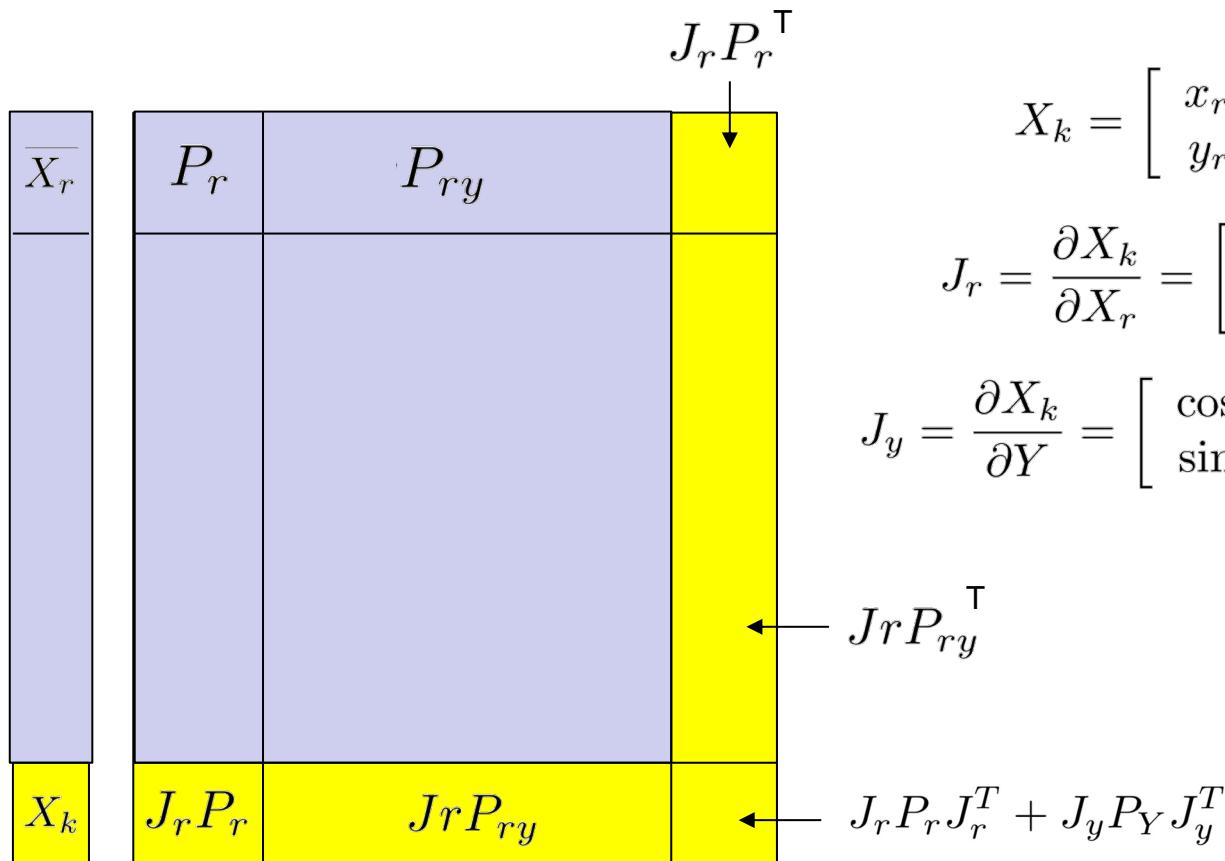
- Pour tout amer perçu
 - S'il est dans la carte : utilisation du filtre pour estimer sa position et celle du robot en fonction des covariances
→ ~ Kalman pour la localisation
 - Si il n'est pas dans la carte : ajout
- Pour savoir s'il est dans la carte
 - amers uniques (caractérisation par les perceptions)
 - distance de Mahalanobis en cas de perceptual aliasing



$$d^2 = \frac{1}{2}(X - Y)^T (P_X + P_Y)^{-1} (X - Y)$$

Ajout des amers

- Agrandir X avec la position absolue
- Agrandir P avec la covariance

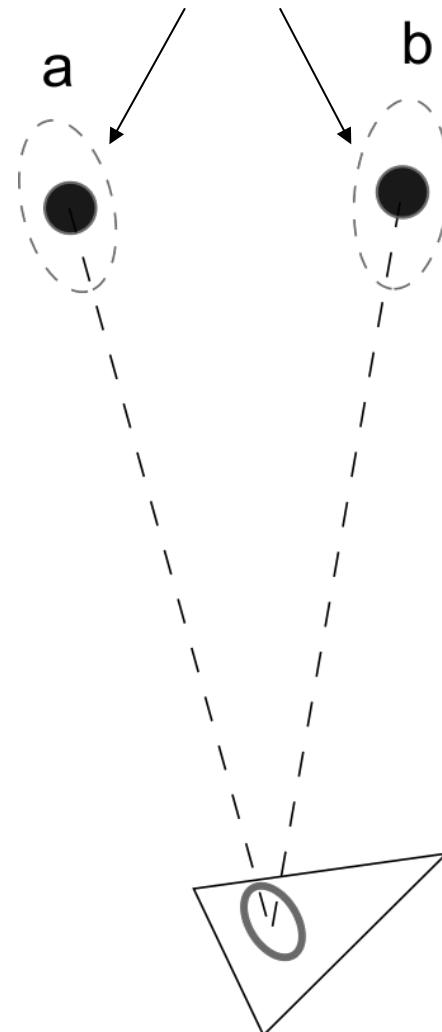


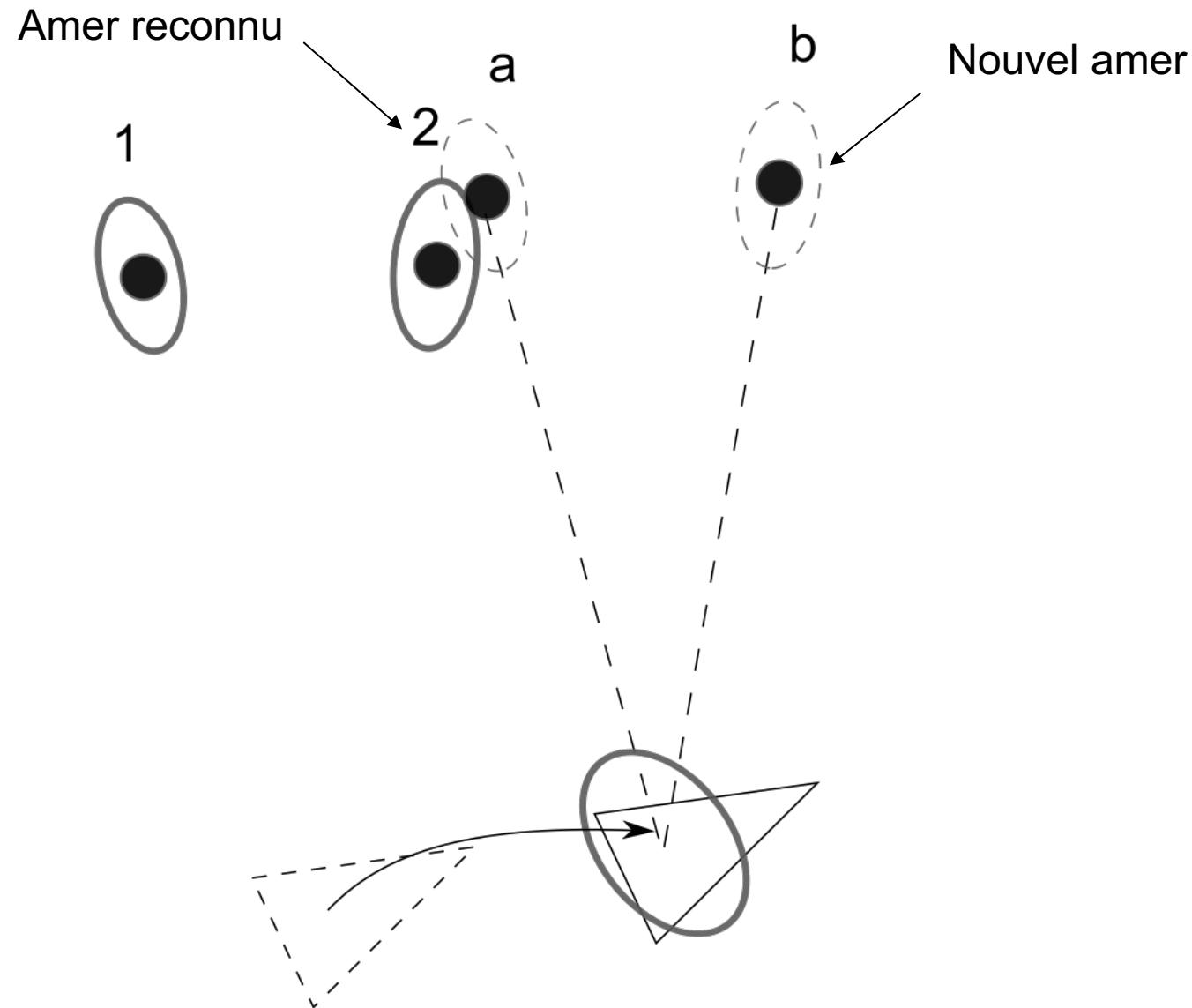
$$X_k = \begin{bmatrix} x_r + d_k \cos(\theta_r + \phi_r) \\ y_r + d_k \sin(\theta_r + \phi_r) \end{bmatrix}$$

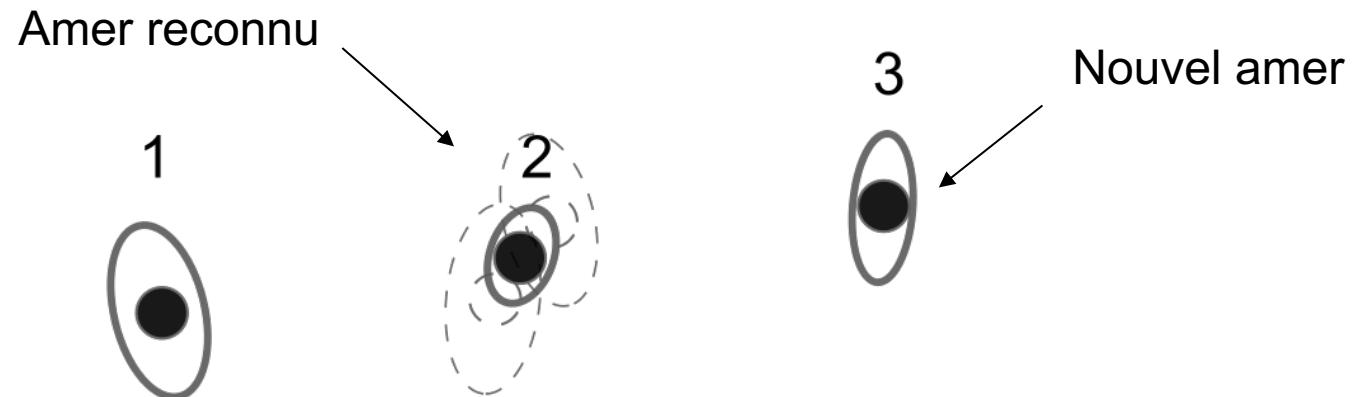
$$J_r = \frac{\partial X_k}{\partial X_r} = \begin{bmatrix} 1 & 0 & -d \sin(\theta + \phi) \\ 0 & 1 & d \cos(\theta + \phi) \end{bmatrix}$$

$$J_y = \frac{\partial X_k}{\partial Y} = \begin{bmatrix} \cos(\theta_r + \phi_r) & -d \sin(\theta + \phi) \\ \sin(\theta_r + \phi_r) & d \cos(\theta + \phi) \end{bmatrix}$$

Nouveaux amers

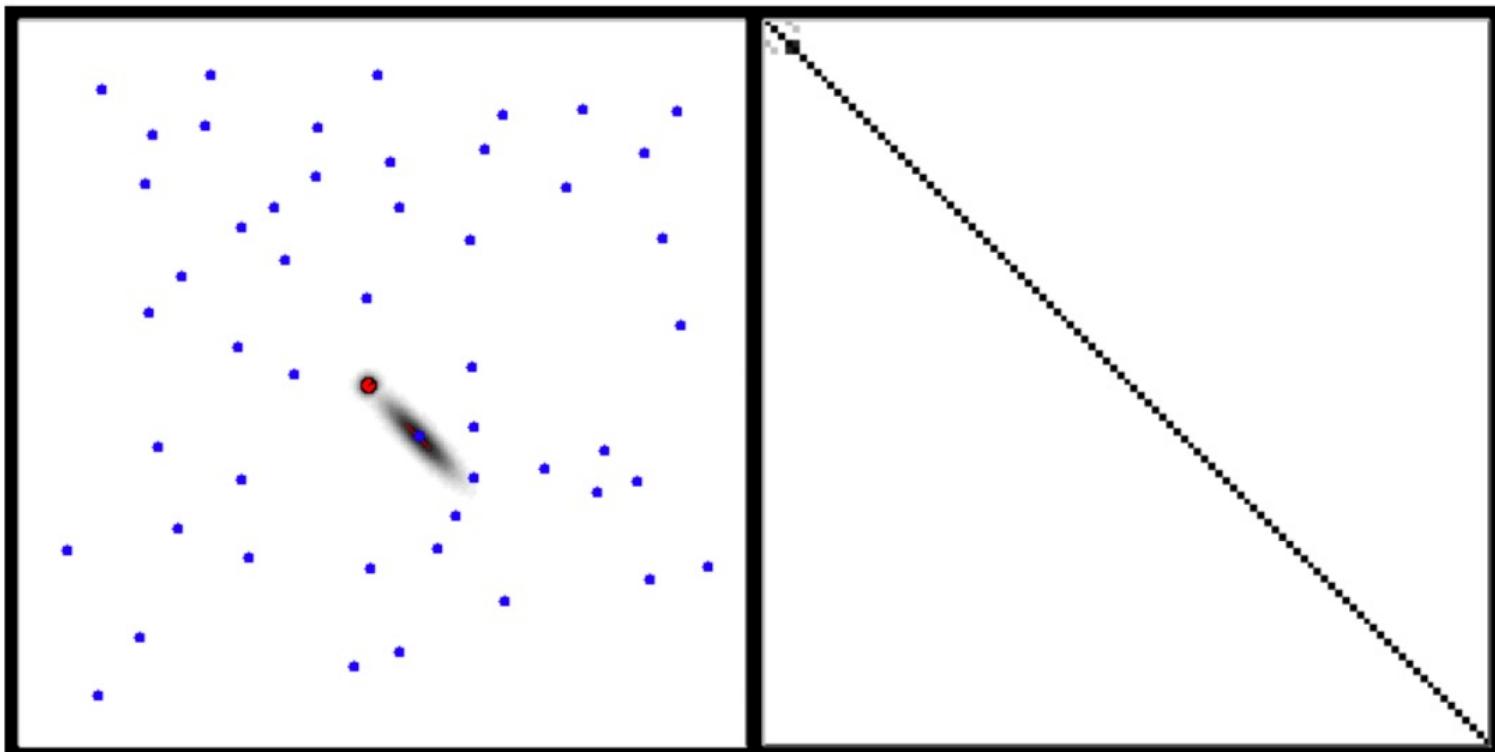






En simulation

(a)

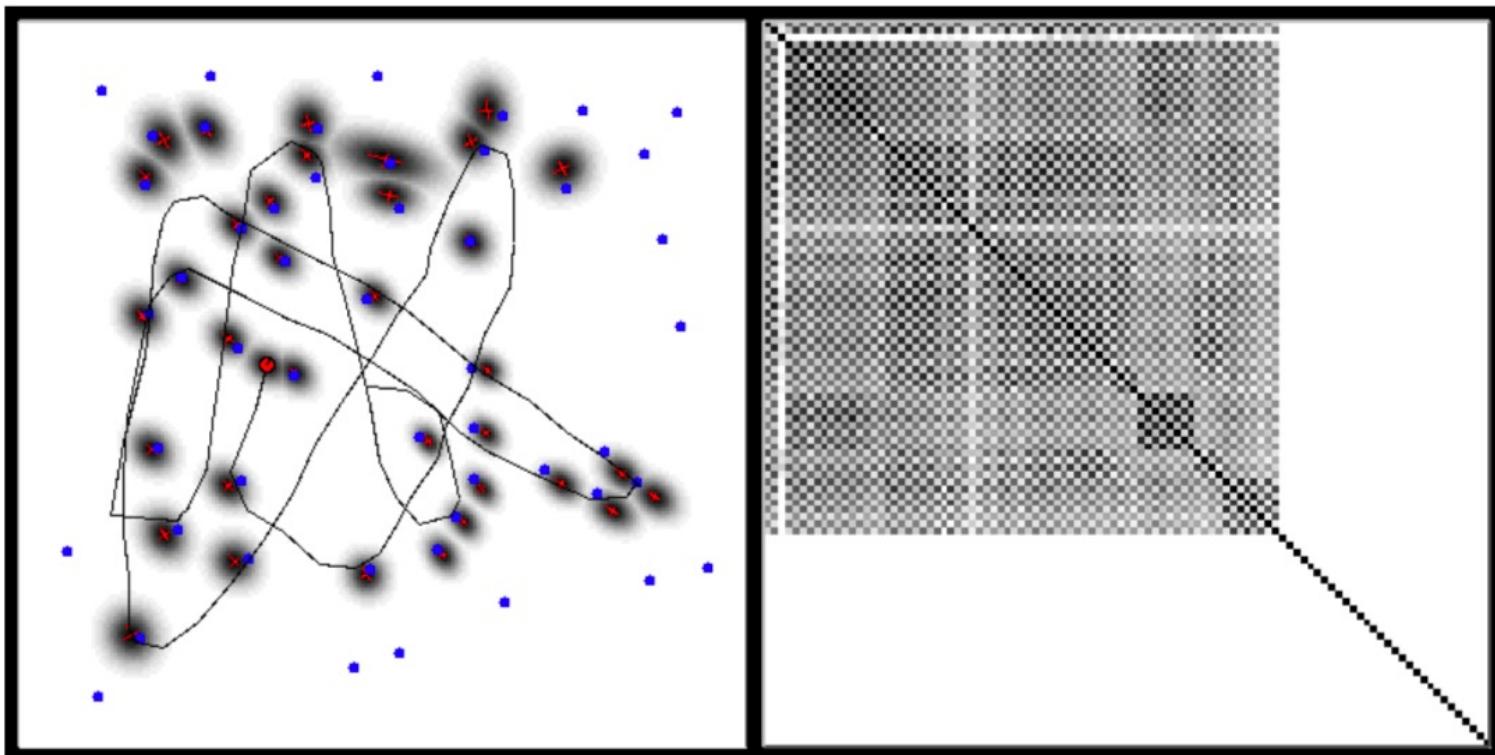


Carte + trajectoire

Matrice de covariance

En simulation

(b)

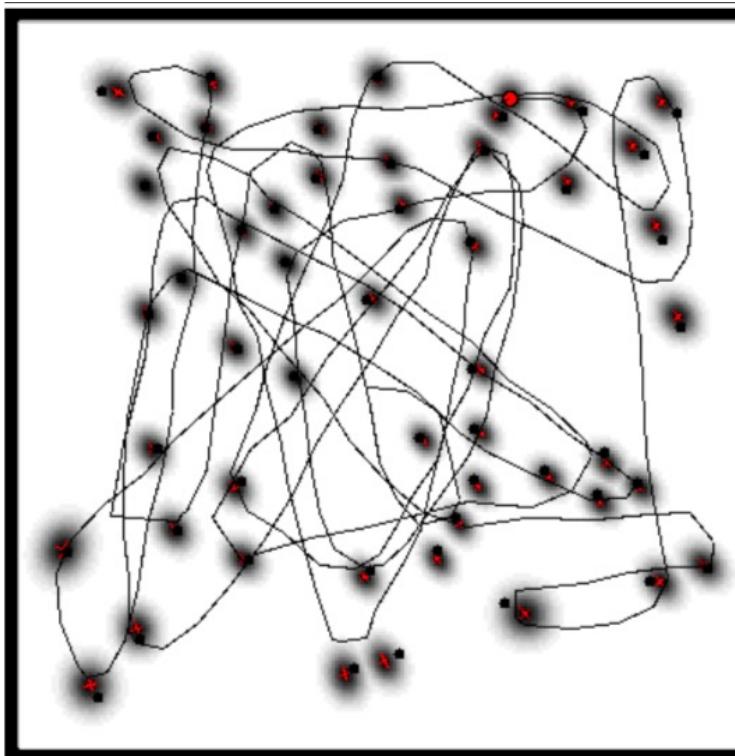


Carte + trajectoire

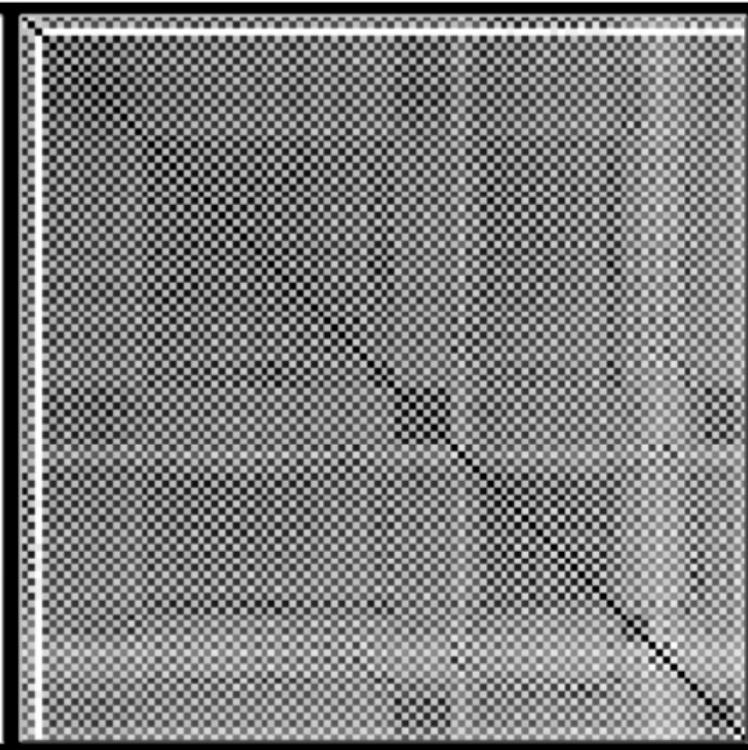
Matrice de covariance

En simulation

(c)



Carte + trajectoire

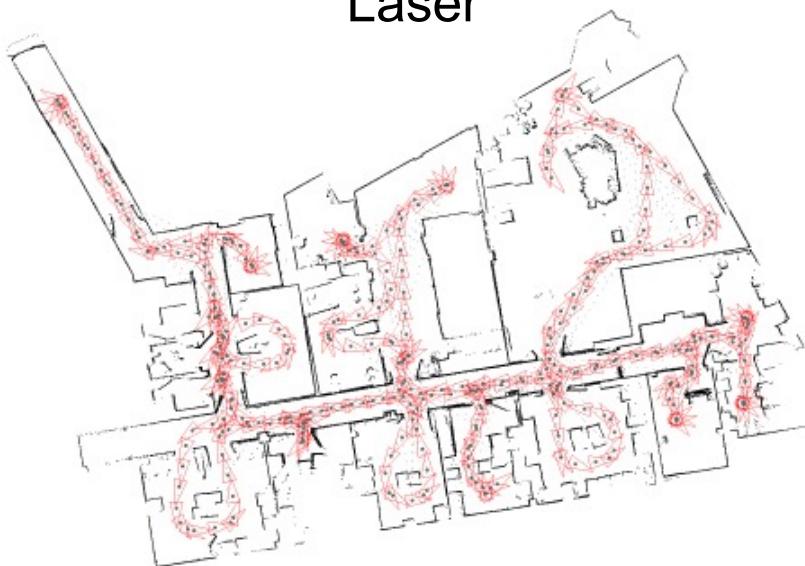
Matrice de covariance
Covariance 'pleine' à la fin

Retour en arrière ?

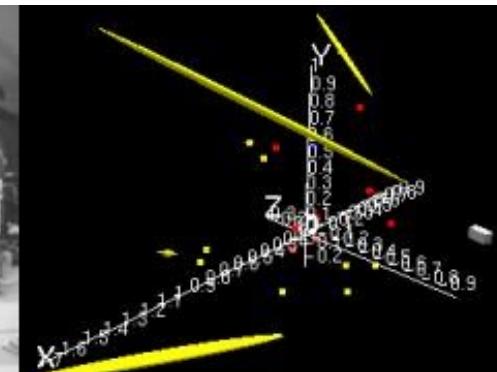
- via la *matrice de covariances*
- Mémorise les relations entre les éléments du vecteur d'états
- Propage de nouvelles estimations aux éléments en relation

Implémentations

Laser



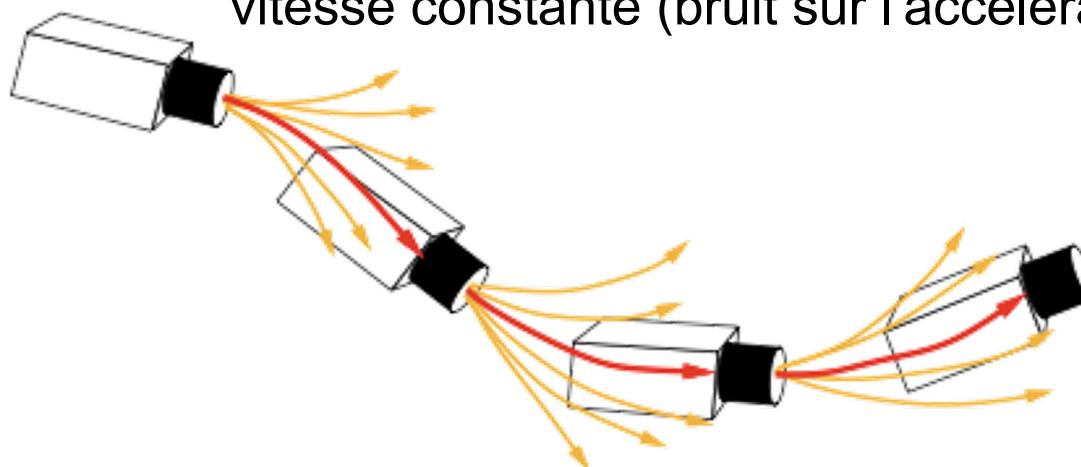
Vision



SLAM 3D avec une caméra monoculaire

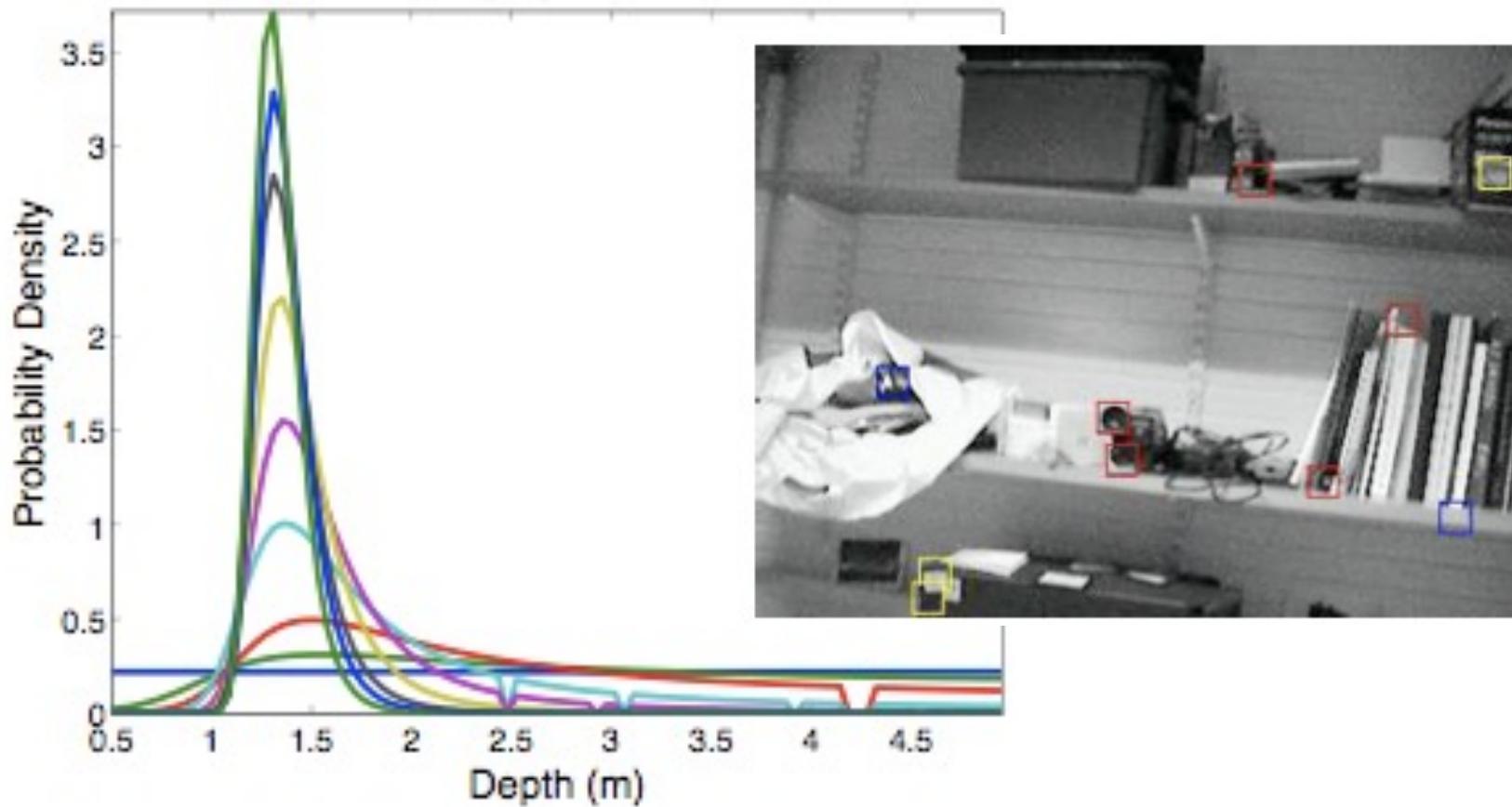
- *MonoSLAM: Real-Time Single Camera Davison. Reid, Molton, and Stasse, PAMI 2007*
- Problème de « structure from motion »
- Etat : position (6 ddl) + vitesses
- Modèle de mouvement classique (Inertie, Odométrie) sur robot
- Modèle de mouvement sans odométrie (caméra seule)
vitesse constante (bruit sur l'accélération)

$$\mathbf{x}_v = \begin{pmatrix} \mathbf{r}^W \\ \mathbf{q}^{WR} \\ \mathbf{v}^W \\ \omega^R \end{pmatrix}$$



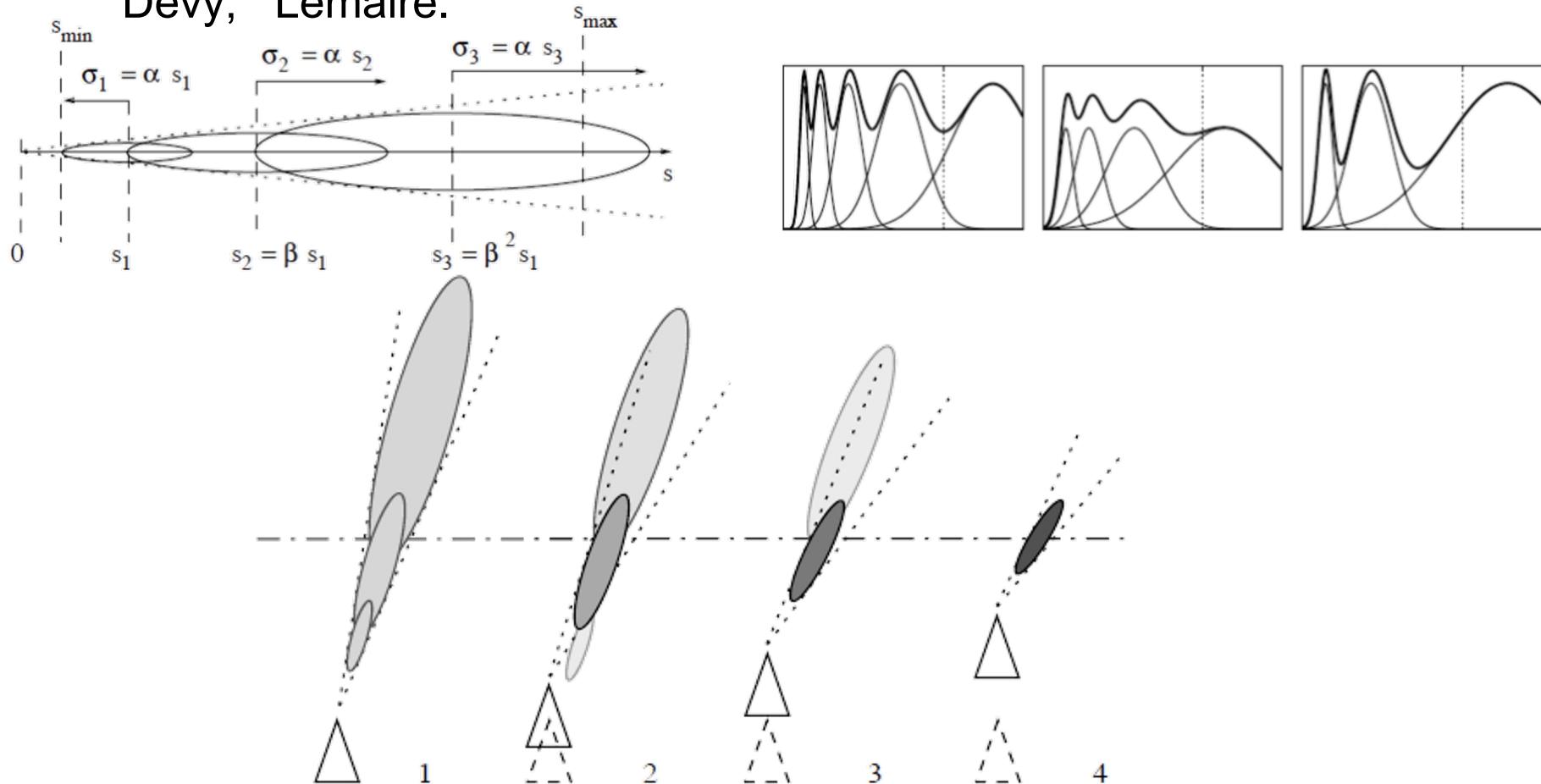
SLAM 3D avec une caméra monoculaire

- Problème d'initialisation des features 3D d'après image
- Initialisation retardée : droite 3D + filtre particulier auxiliaire pour distance



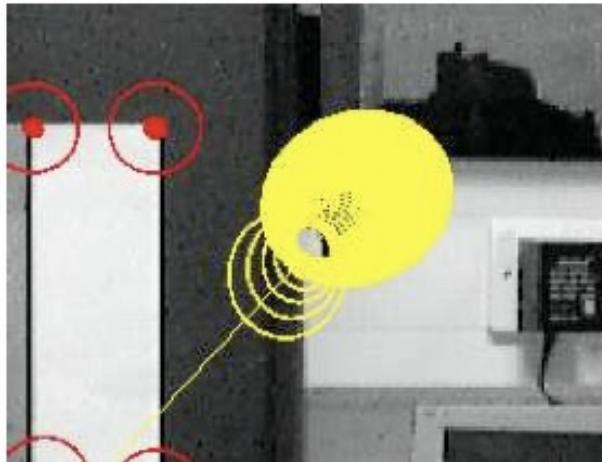
SLAM 3D avec une caméra monoculaire

- Initialisation non retardée : cone \approx multi gaussiennes
- *Undelayed initialization in bearing only SLAM* Sola, Monin, Devy, Lemaire.

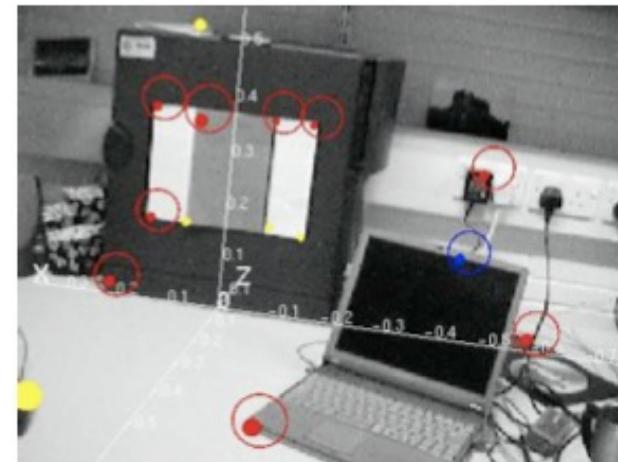


Association de données

- Correspondance carte <-> image
- Recherche active dans l'image



nouvelle feature
(filtre particulaire)

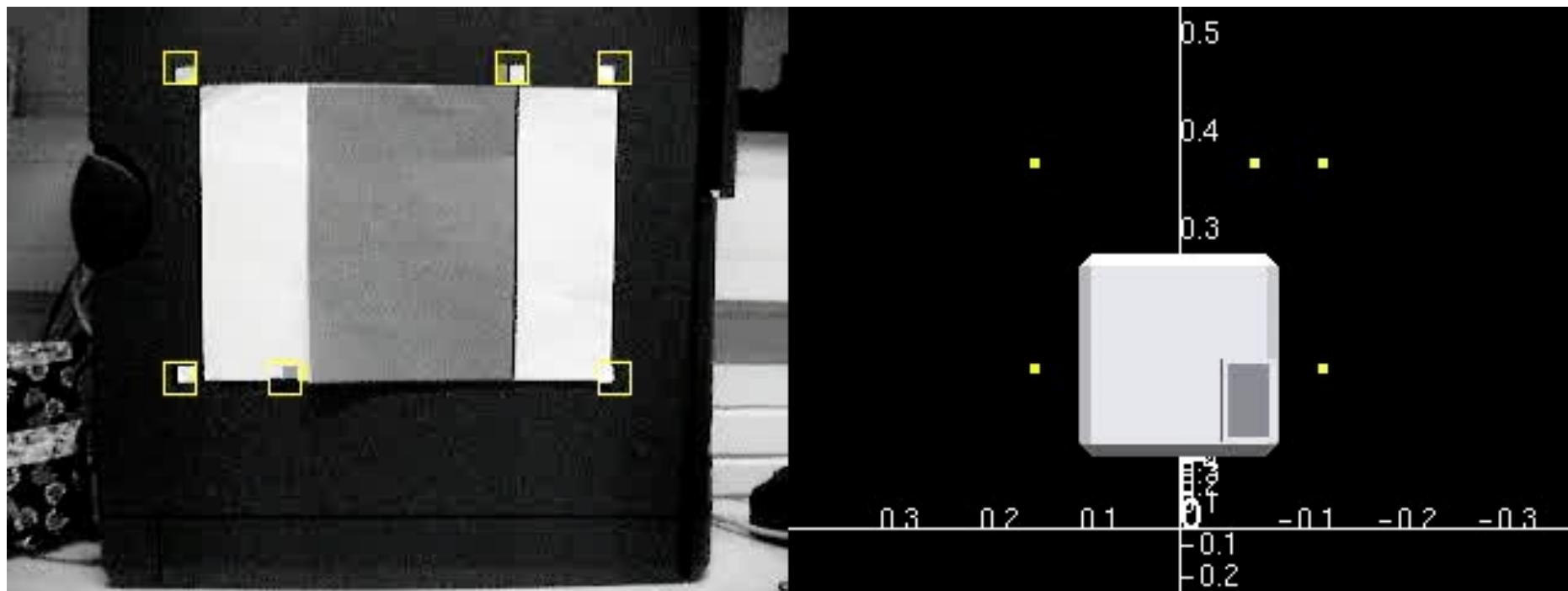


features de la carte

Mise à jour du filtre

- Fonctionne à 30 Hz
- Carte limitée à 100 amers (EKF)

Exemple

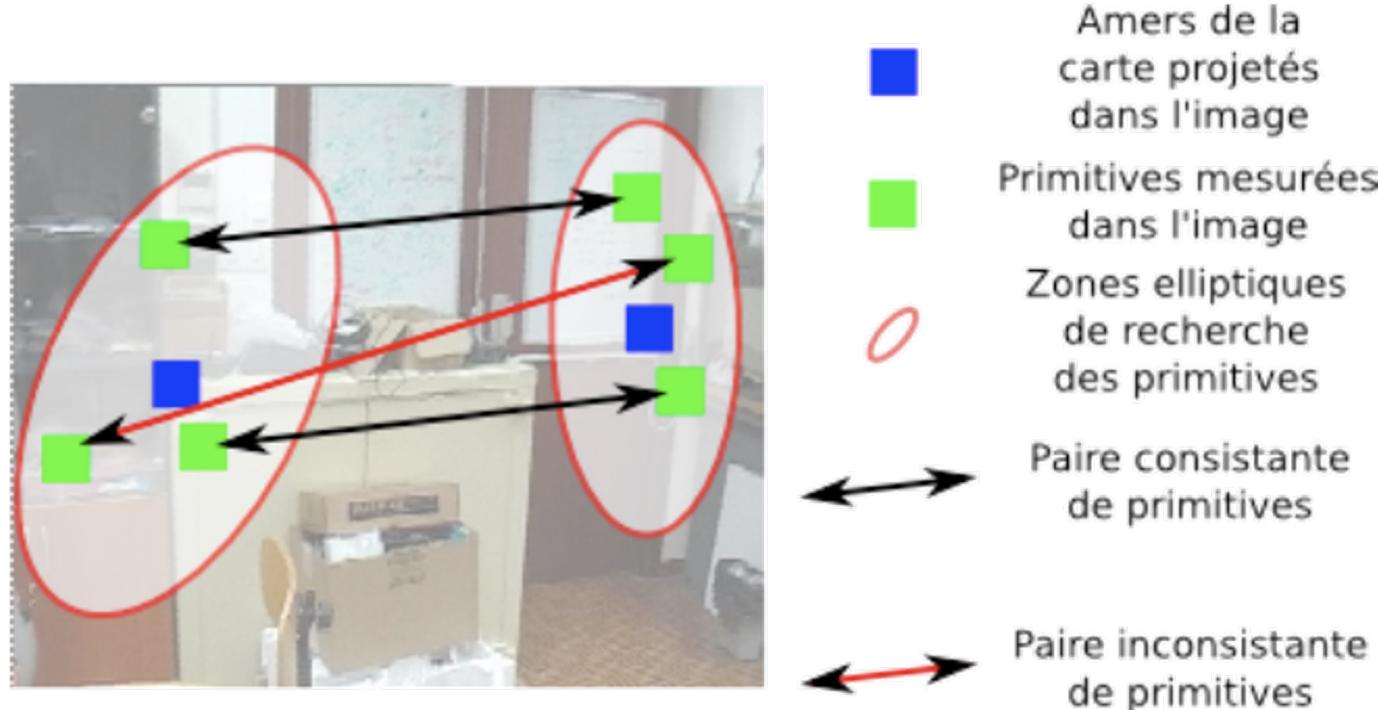


Exemple

Real-Time
Camera Tracking
in Unknown Scenes

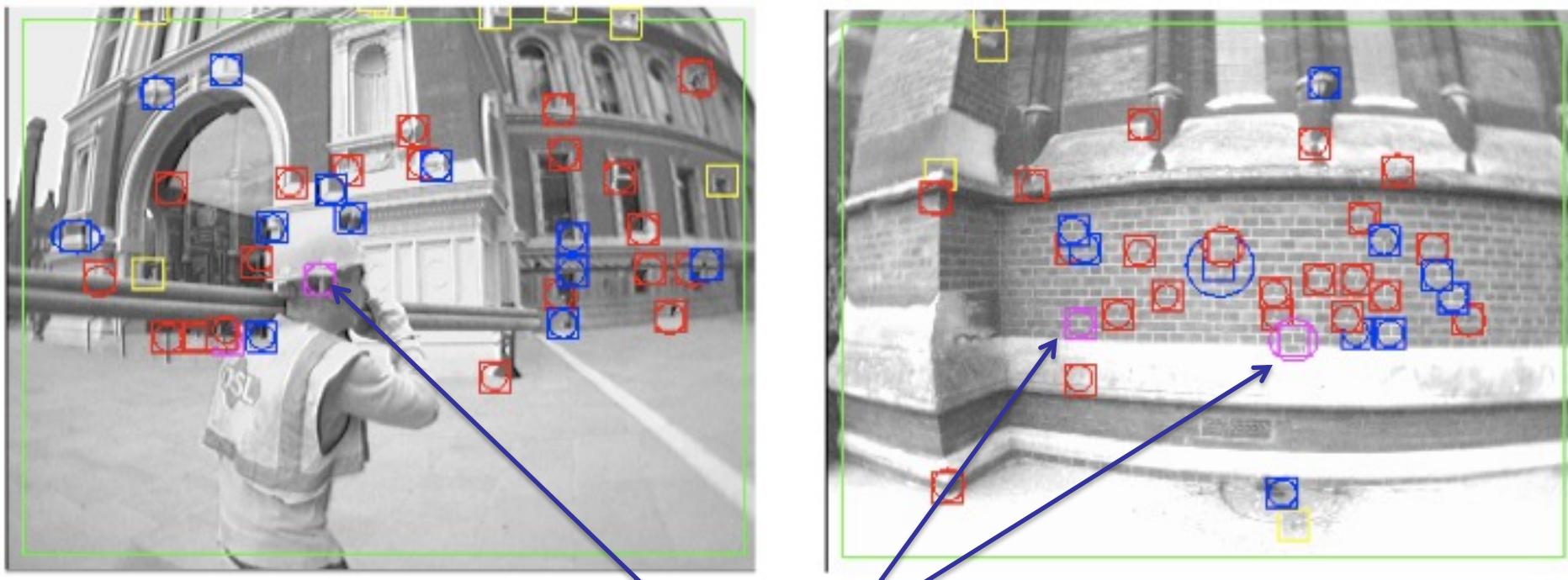
Appariements

- Très sensible aux erreurs d'appariement amer perçu/amer de la carte.
- Une erreur peut entraîner un échec complet (dérive)
 - > Méthodes de matching plus robustes
- Joint Compatibility Test : Mahalanobis sur l'innovation conjointe



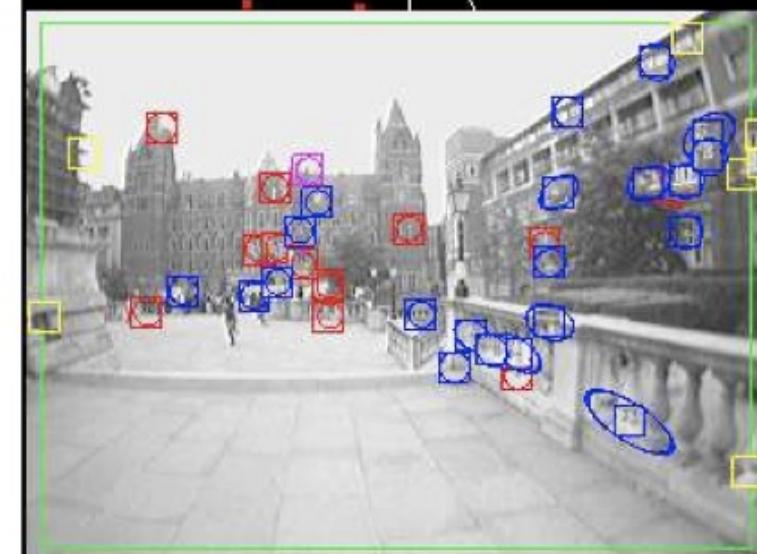
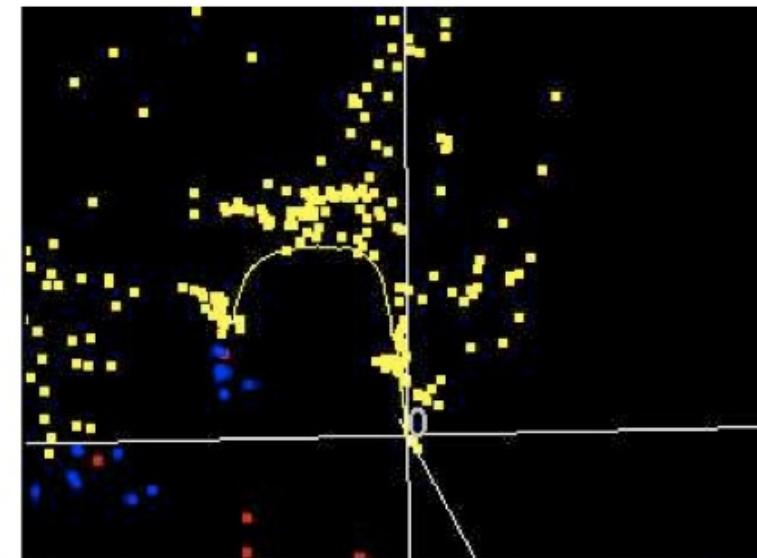
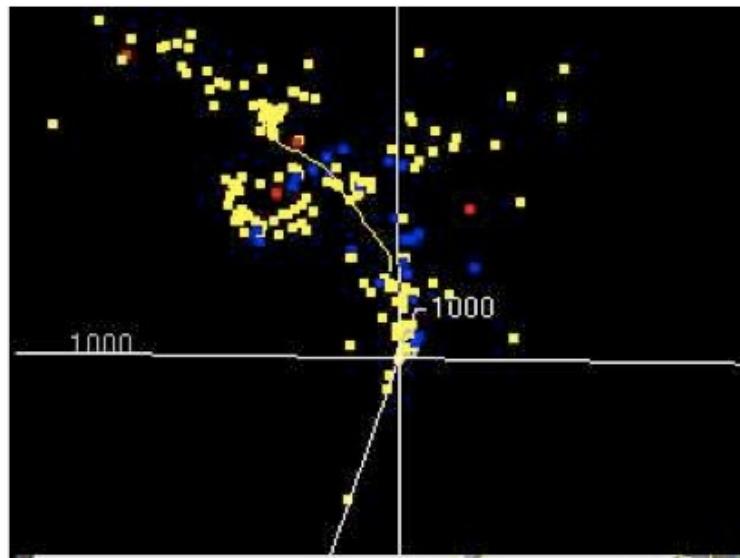
Joint Compatibility Test

- Permet l'appariement de features ambiguës
- Evite le matching de features mobiles



Rejeté

Joint Compatibility Test

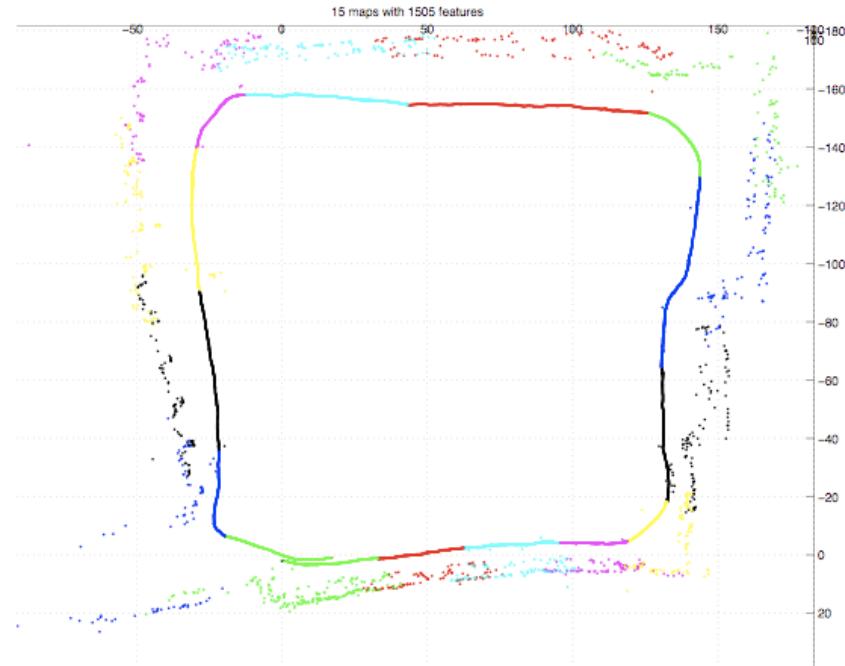
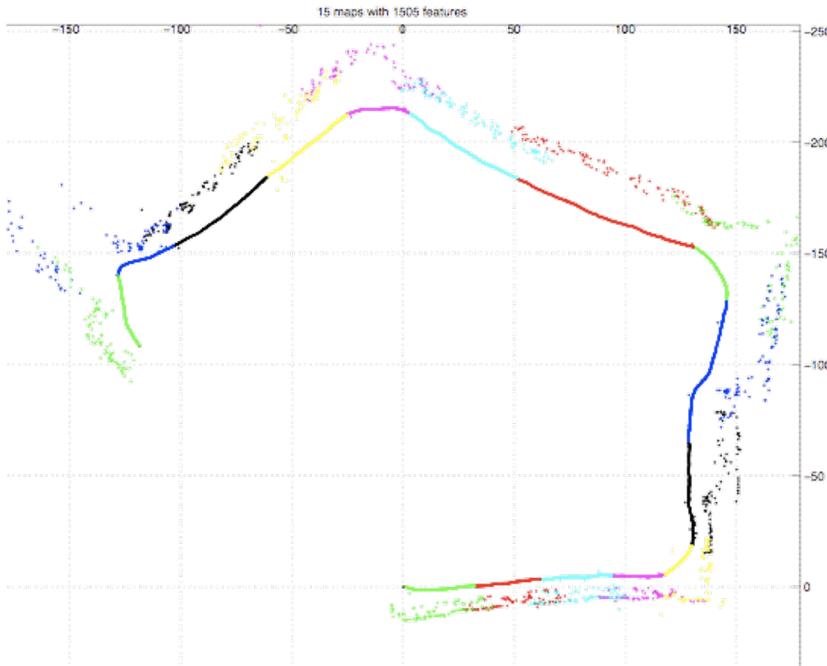


Complexité du SLAM EKF

- Complexité en $O(N^2)$ (matrice de covariance)
 - Négliger les covariances ne fonctionne pas
- Méthodes hiérarchiques
 - Cartes locales
 - Relations entre carte
- Création de sous-cartes
 - Nombre de features > seuil
 - Covariance des features > seuil
 - Structure de l'environnement

Utilisation de cartes locales

- Utilisation de cartes locales de taille M et calcul des covariances dans ces carte $\rightarrow O(M^2)$
- Détection de fermeture de boucle (map matching) + optimisation
- *Mapping Large Loops with a Single Hand-Held Camera* Laura A. Clemente, Andrew J. Davison, Ian Reid, José Neira and Juan D. Tardós, RSS 2007.



TP

- SLAM par filtrage de Kalman

Prochains cours

- SLAM – Partie 2
- TP GraphSLAM