Mini-projet Groupe 3 – Optimisation

MINES ParisTech
http://oasis.mines-paristech.fr *2019-2020*

Ce mini-projet, à effectuer en binôme au sein du même groupe de PC, fera l'objet d'un rapport incluant notamment équations et graphiques obtenus par des simulations sous Python. La forme de ce rapport est laissée libre (pdf, notebook...). Plusieurs rendus intermédiaires sont attendus auprès de votre chargé de PC.

Optimisation de la trajectoire d'un robot

Dans ce sujet, l'objectif est de calculer la trajectoire d'un robot à partir de deux types de mesures : des mesures d'odométrie (représentant les déplacements itératifs d'un robot) et des mesures de fermeture de boucle (lorsque le robot détecte le fait d'être revenu à un endroit déjà visité). La trajectoire est discrétisée en un ensemble fini de poses. On définit alors les différentes poses du robot par une position 2D et un angle représentant la direction du robot.

Lors de son exploration, le robot va mesurer ses différents déplacements au cours du temps : on appelle cela l'odométrie. A certains moments, le robot va repasser par des endroits déjà visités. Lorsque la détection d'un endroit déjà visité se passe correctement, on appelle cela une fermeture de boucle. Nous allons alors créer une relation entre la pose j et une pose déjà connue i. Une transformation est calculée (par l'algorithme de détection de boucle) et est considérée comme connue.

L'ensemble de ces relations peuvent se mettre sous la forme d'un graphe que l'on appelle graphe de poses (voir Figure 1).

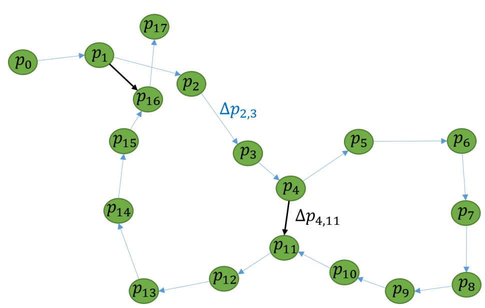

Figure 1 – Exemple d'un graphe de poses avec 18 poses, 17 relations d'odométrie et 2 relations de fermeture de boucle.

1 Modélisation

1. **Formaliser les inconnues du problème.**

   On considère que l'on a en tout $N+1$ poses. Les poses p_i s'écrivent : $p_i = (x_i, y_i, \theta_i)^T$ avec i variant de 0 à N. (x_i, y_i) est la position 2D du robot et θ_i est l'angle en radian représentant la direction du robot (attention à avoir toujours $\theta_i \in]-\pi, \pi]$). Les poses p_i sont les positions du robot dans le repère de la première pose p_0.

   La première pose p_0 est fixée et sert de repère pour toutes les autres poses donc $p_0 = (0, 0, 0)$.

Nous avons donc N poses inconnues ce qui donne la variable $p = (p_1^T, \dots, p_N^T)$ avec $3N$ paramètres en tout.

2. **Trouver la relation liant une pose et la pose suivante à parti d'une transformation relative connue.**

   L'odométrie permet de connaître la transformation relative entre deux poses successives du robot. On considère que la transformation relative $\Delta p_{i,i+1} = (\delta x_{i,i+1}, \delta y_{i,i+1}, \delta\theta_{i,i+1})^T$ est connue et est exprimée dans le repère du robot à la pose i. A partir de cette information, il est donc possible de calculer les poses de façon itérative : $p_{i+1} = R_i \Delta p_{i,i+1} + p_i$ avec la matrice de rotation R_i qui s'écrit sous la forme :

$$R_i = \begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

3. **Formuler la fonction objectif à minimiser permettant de vérifier au mieux les relations d'odométrie et de fermeture de boucle. Justifier votre choix.**

   Les relations d'odométrie nous permettent de calculer une trajectoire initiale. En ajoutant les informations de fermeture de boucle, nous devons trouver une trajectoire qui va minimiser les erreurs pour les relations d'odométrie et les erreurs sur les relations de fermeture de boucle.
   On note L l'ensemble des (i, j) qui sont des fermetures de boucle entre la pose i et la pose j. On considère que l'on connait la transformation relative entre les poses i et j lors d'une fermeture de boucle : $\Delta p_{i,j} = (\delta x_{i,j}, \delta y_{i,j}, \delta\theta_{i,j})^T$ (qui est aussi exprimée dans le repère de la pose i).

   Pour trouver la trajectoire optimale, on cherche à minimiser la fonction f :

$$f(p) = \sum_{i=0}^{N-1} ||p_{i+1} - R_i \Delta p_{i,i+1} - p_i||_2^2 + \gamma \sum_{(i,j)\in L} ||p_j - R_i \Delta p_{i,j} - p_i||_2^2$$

   On peut aussi écrire f sous la forme équivalente :

$$\begin{aligned} f(p) &= f(x_1, y_1, \theta_1, x_2, y_2, \theta_2 \dots, x_N, y_N, \theta_N) \\ &= \sum_{i=0}^{N-1} \Big[(x_{i+1} - \cos(\theta_i)\delta x_{i,i+1} + \sin(\theta_i)\delta y_{i,i+1} - x_i)^2 + (y_{i+1} - \sin(\theta_i)\delta x_{i,i+1} - \cos(\theta_i)\delta y_{i,i+1} - y_i)^2 \\ &\qquad + (\theta_{i+1} - \delta\theta_{i,i+1} - \theta_i)^2\Big] \\ &\quad + \gamma \sum_{(i,j)\in L} \Big[(x_j - \cos(\theta_i)\delta x_{i,j} + \sin(\theta_i)\delta y_{i,j} - x_i)^2 + (y_j - \sin(\theta_i)\delta x_{i,j} - \cos(\theta_i)\delta y_{i,j} - y_i)^2 \\ &\qquad + (\theta_j - \delta\theta_{i,j} - \theta_i)^2\Big] \end{aligned}$$

   Il s'agit d'un problème de moindre carrés non linéaires dont on cherche le minimum avec $3N$ paramètres et $3N + 3 * Card(L)$ résidus.
   γ est un paramètre permettant d'ajuster le poids des informations de fermeture de boucle par rapport aux informations d'odométrie (on peut tout d'abord regarder les solutions avec $\gamma = 1$).

2 Etude et résolution numérique

Pour ce projet, nous allons travailler sur un jeu de données appelé INTEL. Il s'agit d'optimiser la trajectoire d'un robot qui a exploré le laboratoire de recherche d'INTEL (la Figure 2 montre une carte de l'environnement).

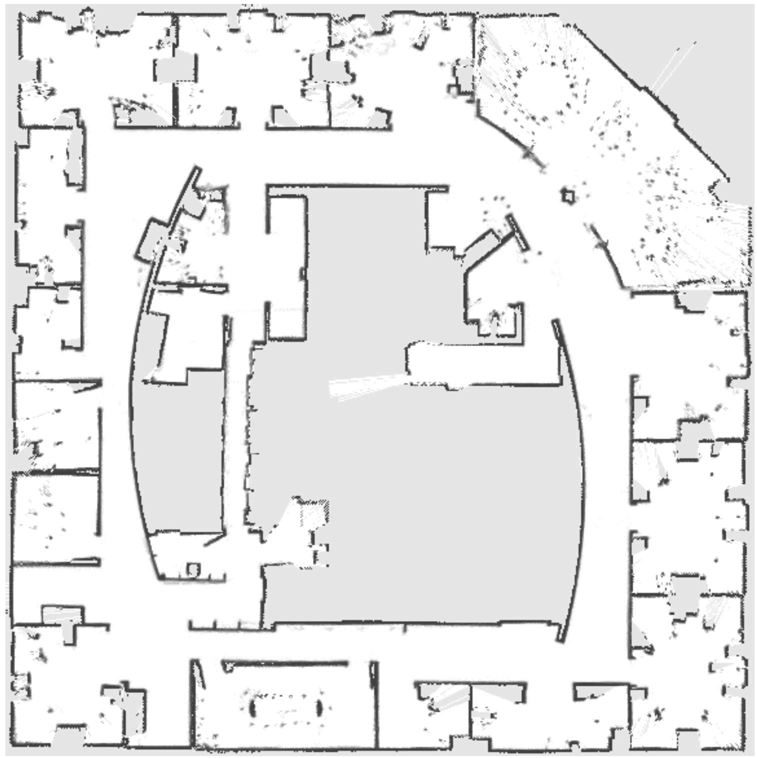

FIGURE 2 – Carte du laboratoire de recherche d'INTEL.

1. Récupérer le fichier "INTEL.txt" à l'URL suivante : `https://cloud.mines-paristech.fr/index.php/s/MrmDh4fkQOXKXOd`.
   Il contient les données d'odométrie et de fermeture de boucle. Il s'agit d'un fichier sous format ASCII avec deux types de ligne :
   `EDGE_ODO i i+1 dx dy dtheta` où $\Delta p_{i,i+1} = (\mathtt{dx}, \mathtt{dy}, \mathtt{dtheta})^T$ (relation d'odométrie)
   `EDGE_LOOP i j dx dy dtheta` où $\Delta p_{i,j} = (\mathtt{dx}, \mathtt{dy}, \mathtt{dtheta})^T$ (relation de fermeture de boucle)
   Implémenter une fonction de lecture de fichier pour récupérer les données du graphe.
2. A partir des données d'odométrie, calculer une première estimation de la trajectoire du robot. Utiliser Matplotlib pour afficher le graphe. Commenter les résultats.
3. Etudier le problème d'optimisation (convexité, conditionnement, existence et unicité d'une solution) par la méthode, analytique ou numérique, de votre choix.
4. Développer un algorithme de résolution du problème. Justifier votre choix. Afficher le graphe optimisé. Commenter les résultats obtenus (en comparant votre trajectoire avec la carte de la Figure 2).

3 Etude avancée

1. Changer le produit scalaire utilisé dans ce problème pour y associer une matrice symétrique définie positive (cad dans la formulation vectorielle de f, on remplace la norme $||x||_2^2 = x^T x$ par la norme $||x||_\Omega^2 = x^T \Omega x$ avec Ω matrice symétrique définie positive). Justifier l'intérêt. Pour la suite, prenez Ω diagonale.
2. Modifier votre algorithme en conséquence et tester le sur le dataset suivant "MIT.txt" : `https://cloud.mines-paristech.fr/index.php/s/OZcYrTk3C3ft9J1` Commenter les résultats obtenus (en comparant votre trajectoire avec la carte de la Figure 3).

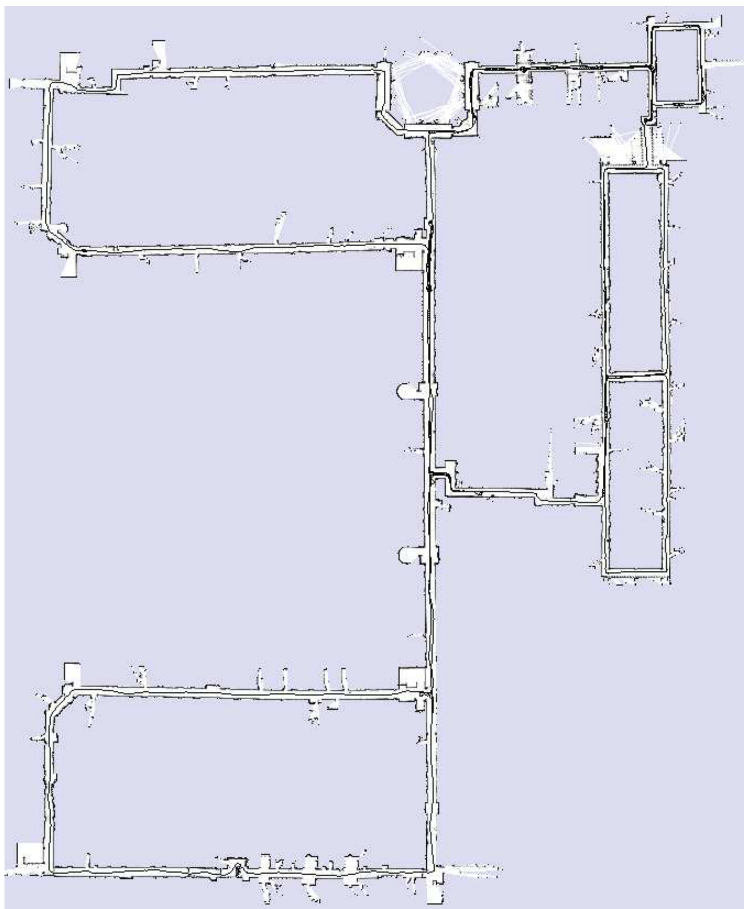

FIGURE 3 – Carte du MIT Killian Court.