



Rapport de TX52

Exploitation de l'éclairage urbain pour localiser et suivre les déplacements d'un véhicule

Hadrien Rothéa
Enseignant responsable : Dr. Nathan Crombez
P21

| | |
|---|-----------|
| Introduction | 2 |
| Etat de l'art | 2 |
| Estimation de position sur carte virtuel existante | 2 |
| SLAM | 3 |
| Estimation de mouvement | 3 |
| Notre approche | 4 |
| Etude sur la projection de caméra omnidirectionnelle | 4 |
| Localisation | 6 |
| Environnement | 6 |
| Traitement d'image | 7 |
| Tracking | 10 |
| Matching et calcul de pose | 11 |
| Etude et résultats | 13 |
| Conclusion et ouverture | 16 |
| Références | 17 |

Introduction

Les systèmes de géolocalisation rapide et à très haute précision sont une technologie de plus en plus demandée. Elle permettrait de grandes avancées dans des domaines comme la voiture autonome. Aujourd'hui, le système gps le plus abouti et performant est la géolocalisation par satellite (GNSS). Celui-ci rencontre cependant quelques difficultés dans des zones urbaines. On appelle ce phénomène le urban canyon. Il est causé par les hauts bâtiments qui bloquent (Non-Line-of-Sight : NLOS) ou reflètent (multipath) le signal.

Pour répondre à cette problématique, une nouvelle approche a été proposée et testée dans cette étude. Celle-ci se base sur la détection d'éclairage de lampadaire par caméra omnidirectionnelle. En effet, les lampadaires se trouvent en abondance dans les milieux urbains et la caméra omnidirectionnelle permet d'obtenir un maximum d'informations. La limite pratique de cette approche est que dans le contexte présent, elle ne serait utilisable que la nuit. Cependant, on anticipe un futur où la lifi (light fidelity) est déployée de partout. La lifi est une technologie de communication par lumière. Elle présente encore des inconvénients mais est extrêmement rapide. Donc dans un contexte futur où la lifi est développée, ce système sera rajouté comme un complément à celle-ci et sera disponible tout au long de la journée.

Etat de l'art

Avant d'aller plus loin dans l'implémentation de la méthode de cette étude nous allons déjà explorer l'état de l'art, qui se rapproche de notre méthode dans le domaine.

Estimation de position sur carte virtuel existante

Cette approche a été testée par Y. Kusuno et al [1] et T. Cham et al [2]. Dans leurs études, ils utilisent une carte virtuelle déjà existante et des images de caméra

omnidirectionnelle. Pour se localiser dans l'espace ils font du matching de patterns entre les patterns trouvés dans l'image 2D de la caméra et ceux présent dans la carte virtuelle. Les algorithmes diffèrent mais les cartes virtuelles aussi. Dans l'étude de Y. Kusuno[1] tous les contours sont stockés et utilisés dans le matching. Et dans l'étude de T.Cham[2] seulement les bords verticaux des bâtiments sont utilisés.

Ces deux approches présentent des résultats encourageants. On peut tout de même soulever certains désavantages. Il y a tout d'abord le grand espace de stockage que doit demander la carte virtuelle pour pouvoir stocker autant d'informations. Les calculs sont aussi lourds car dans l'étude de Y. Kusuno, il est envisagé de faire passer ces calculs côté serveur.

SLAM

SLAM signifie Simultaneous Localization And Mapping. Cette technique consiste donc à cartographier l'environnement et en parallèle se localiser dedans. On peut voir un tel exemple dans l'étude de Xilin Ke et al [3] où ils utilisent une caméra omnidirectionnelle avec cette technique. Les résultats ici aussi sont encourageants.

Cette technique présente l'avantage de ne pas avoir besoin de carte préexistante et remet à jour la carte régulièrement. Cependant, le processus reste lourd avec la cartographie et le calcul de pose en simultané.

Estimation de mouvement

Cette approche propose d'utiliser (en complément à d'autres techniques) des séquences d'images de la caméra omnidirectionnelle pour évaluer le mouvement/la motion. Dans l'étude de K. Khoshelham et al [4], ils utilisent cette technique pour calculer la nouvelle position en fonction de ce mouvement et de la position précédente.

Cette technique présente l'avantage de ne pas avoir besoin de carte virtuelle de l'environnement et d'alléger les calculs. Elle a d'un autre côté de gros désavantages d'accumulation d'erreurs et donc de pertes de précision.

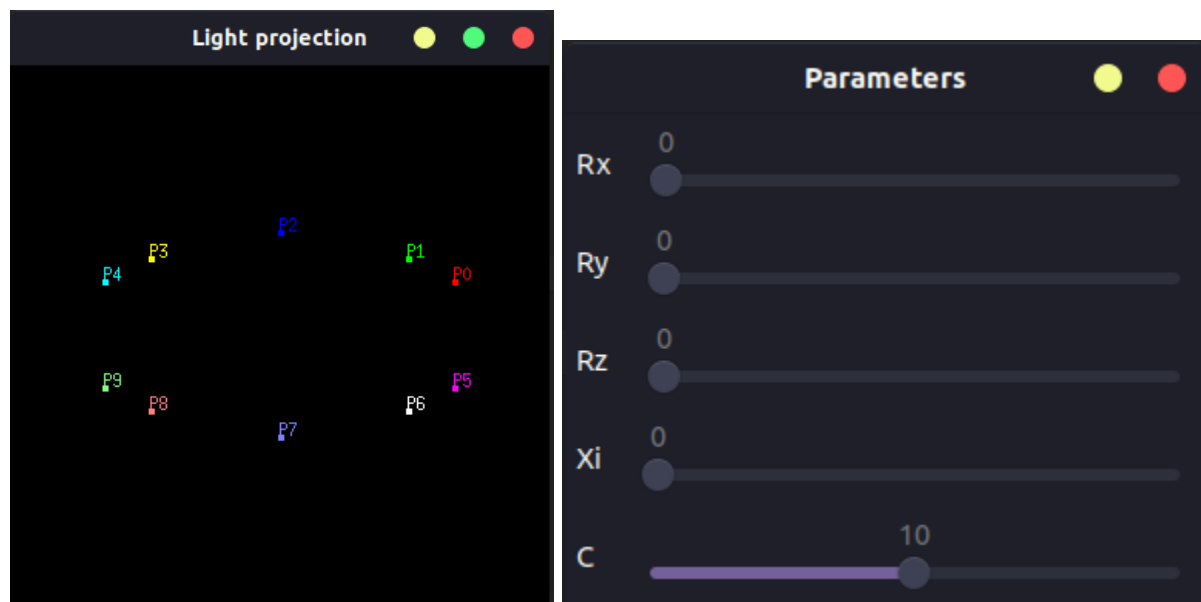
Notre approche

Nous n'avons pas pu trouver une étude présentant une technique plus similaire à l'approche de cette étude. Dans celle-ci, l'on retrouve une carte virtuelle en nuage de points pré-cartographié et une localisation par estimation de pose par points. En théorie cette approche peut jouir de la précision d'autre approche d'estimation de pose et d'une réduction considérable du stockage et des calculs nécessaires.

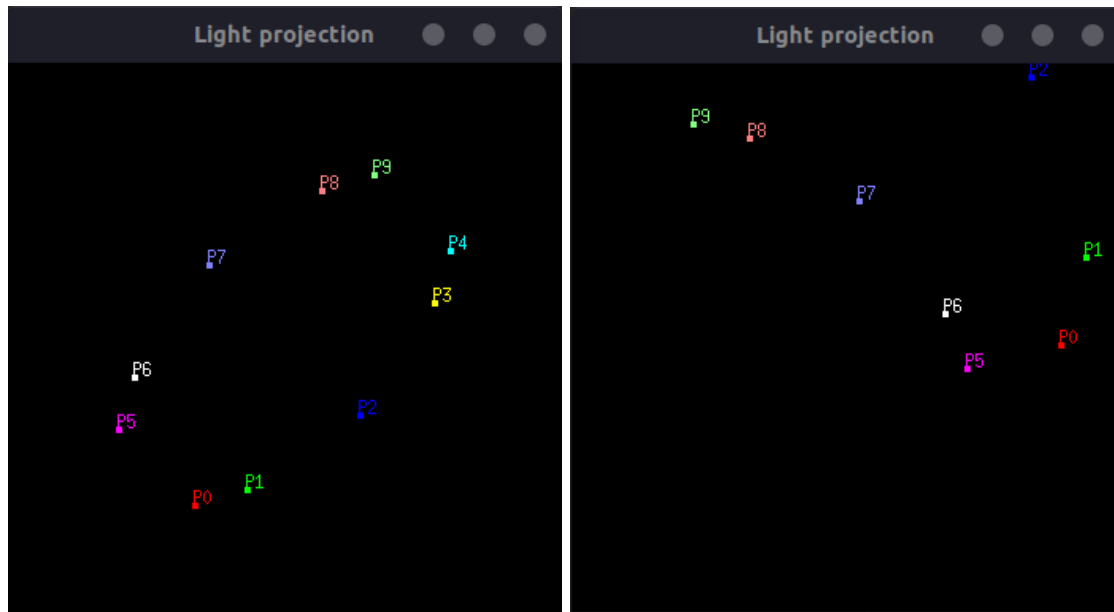
Etude sur la projection de caméra omnidirectionnelle

Premièrement, il a fallu comprendre les mécanismes et physiques de la projection omnidirectionnelle. Pour cela nous avons d'abord appris par le biais de la thèse de N. Crombez [5]. Nous sommes ensuite passé par le développement d'un programme qui permet de tester et visualiser cette projection. Pour ce programme, nous avons utilisé les bibliothèques c++, visp et opencv.

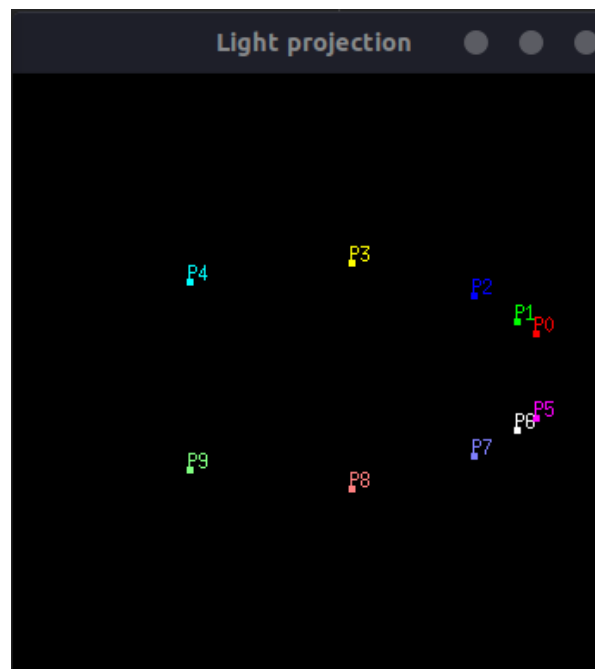
Dans ce programme, on a tout d'abord une projection de 10 points par défaut, ainsi qu'un menu pour changer les différents paramètres.



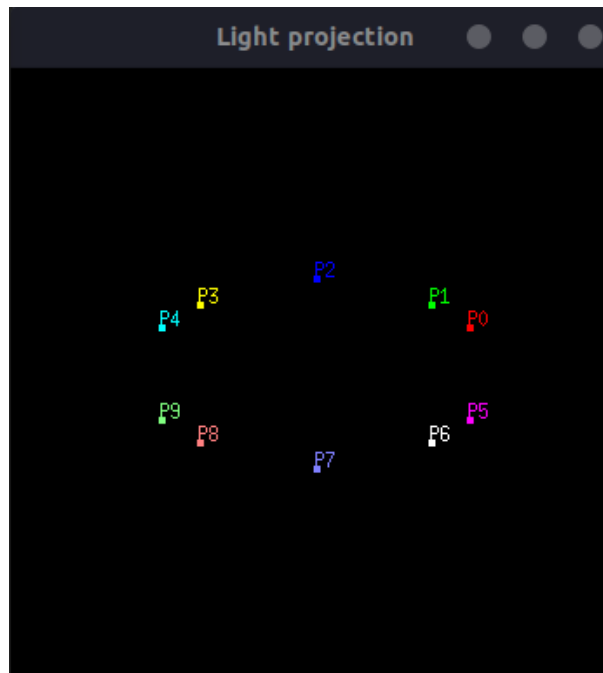
Ensuite on peut effectuer des rotations: (rotation sur l'axe z de $\pi/4$ & rotation de $2\pi/6$ sur x + $\pi/4$ sur z)



On peut aussi faire des translations sur x :



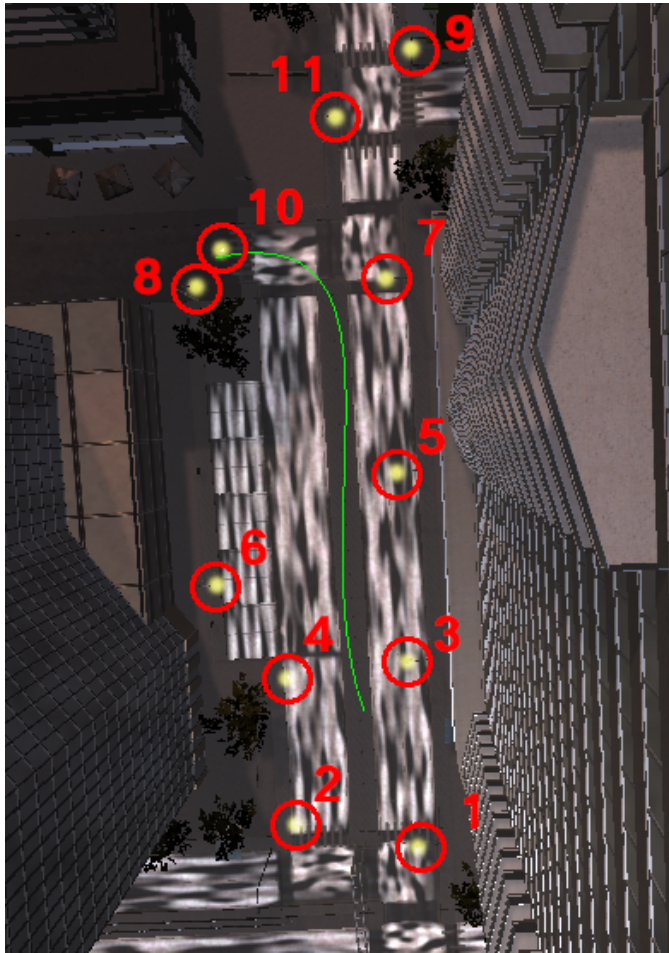
Et enfin on peut changer la valeur xi de la caméra :



Localisation

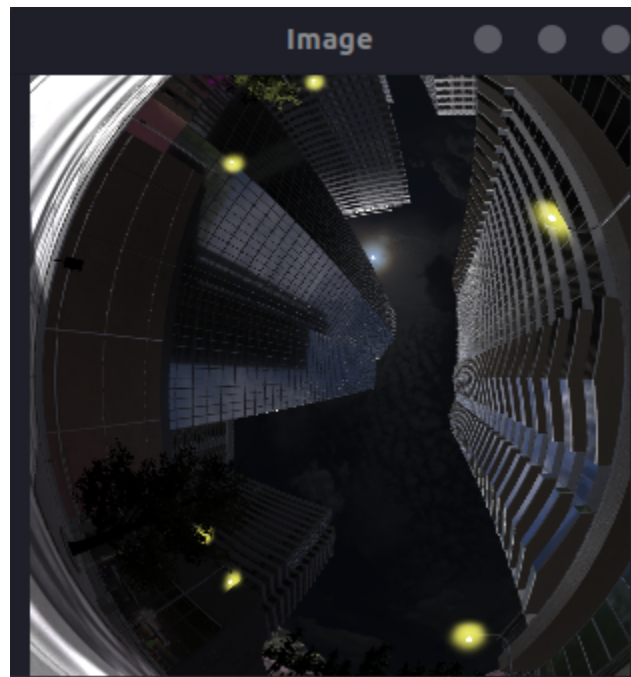
Environnement

Tout ce qui a été fait dans cette étude a été réalisé dans un environnement virtuel donc ne représente pas toutes les contraintes que l'on rencontrerait dans le monde réel. Ce dont dispose notre programme, ce sont les frames d'une avancé dans un milieux urbain ainsi que la pose initiale de la caméra et les coordonnées 3D de tous les lampadaires. Notre carte se présente sous la forme de nuages de points 3D avec le point situé au niveau de la lampe du lampadaire.

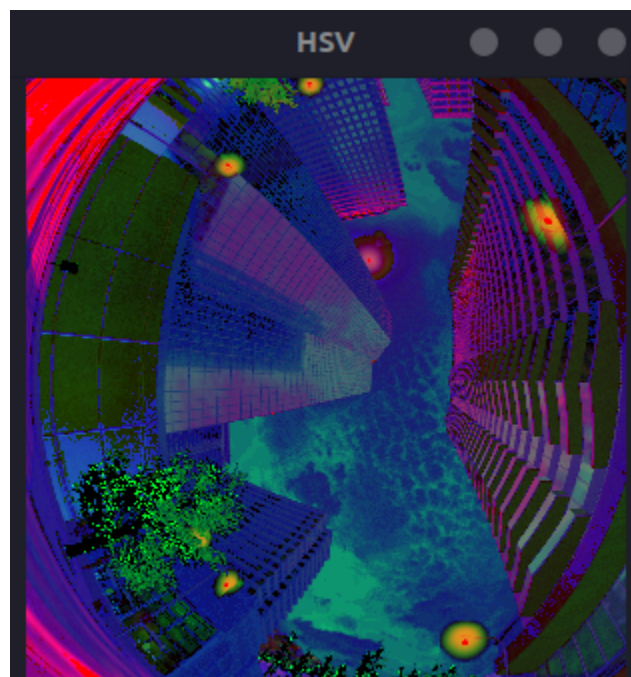


Traitement d'image

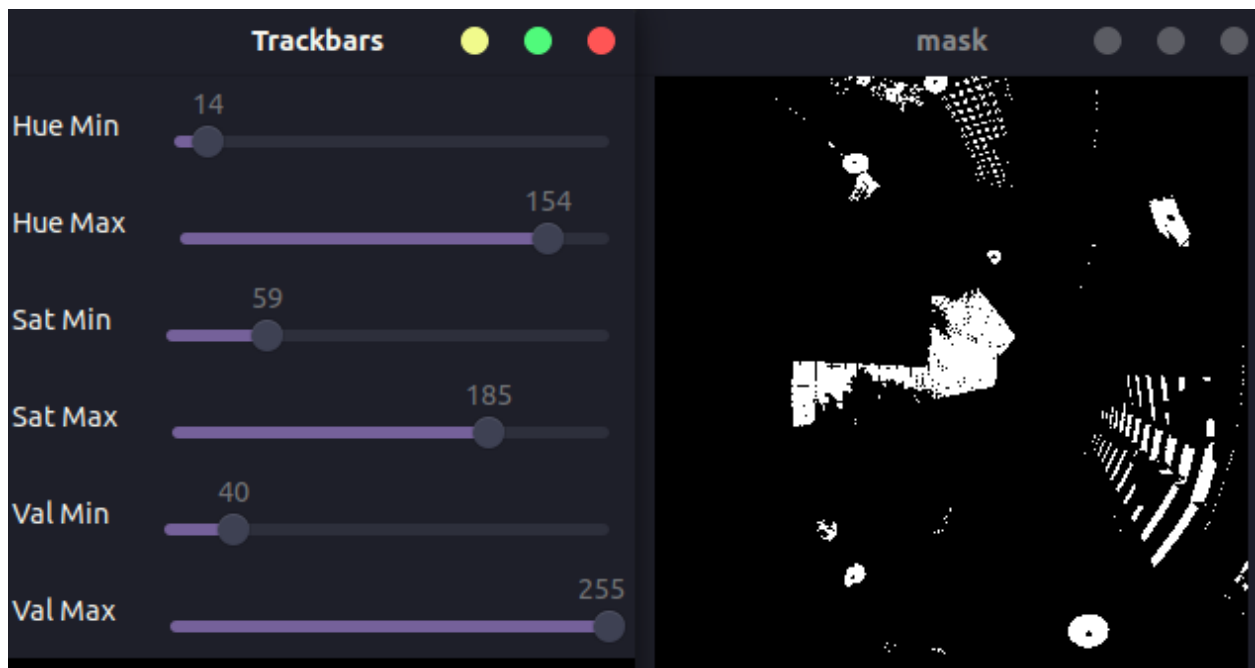
Voici la représentation d'une frame dans notre environnement :



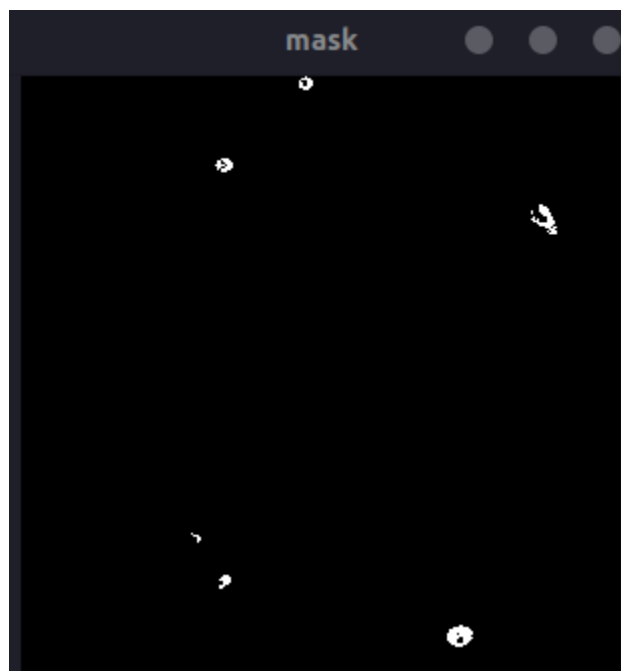
De cette image nous voulons récupérer toutes les coordonnées 2D des lampadaires. Pour cela, nous sommes partis sur une détection lumineuse très basique. Nous commençons par convertir l'image en HSV.



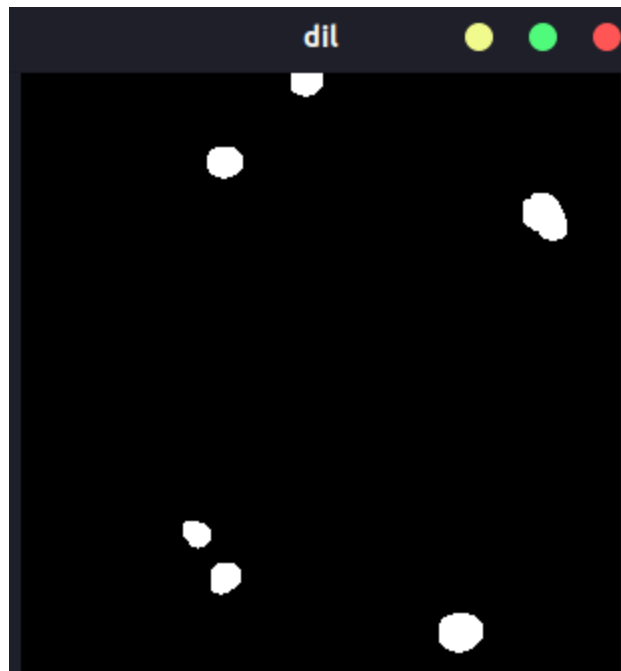
Ensuite, nous créons un mask en paramétrant les Hmax/Hmin, Smax/Smin et Vmax/Vmin.



Nous avons créé un petit programme en parallèle qui permet de modifier ces paramètres et de changer en direct le mask. Ainsi nous avons pu trouver les paramètres optimaux pour ne détecter que des lampadaires (en blanc) :



Pour mieux détecter et unifier les lampes, nous avons enfin appliqué une dilatation des points blancs :



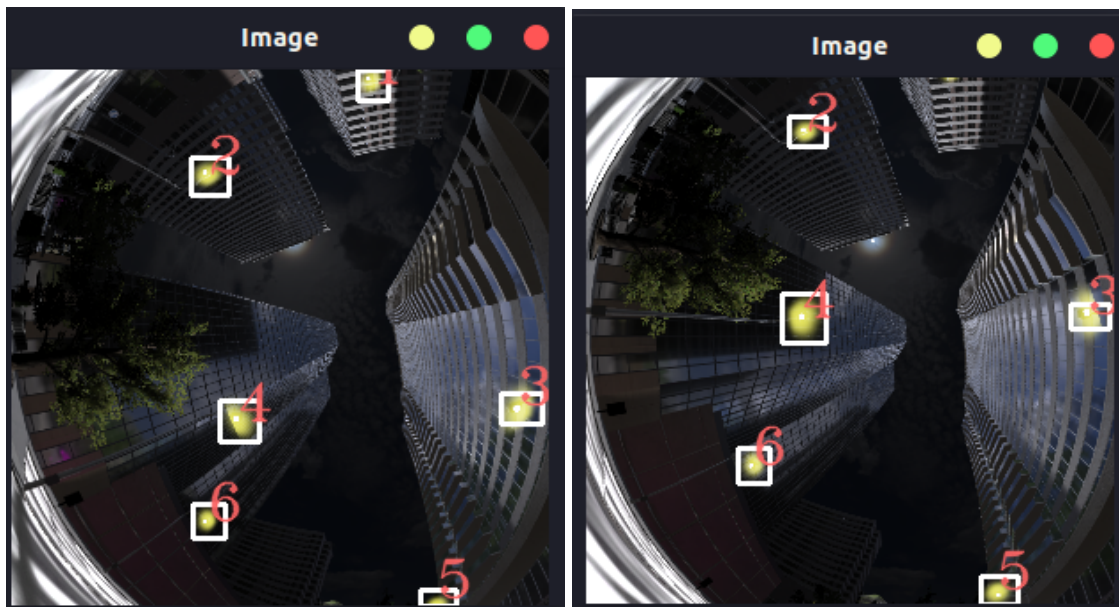
Enfin pour récupérer les coordonnées 2D des lampadaires nous prenons le point central de chaque blobs.

Cette technique permet d'obtenir de très bon résultats et ne détecte pas des bruits comme la lune et les reflections. Il est tout de même important de noter que ceci marche sur notre simulation et non un environnement réel où d'autres bruits comme des feux de voitures ou des lumières dans les bâtiments rendrait cette approche plus difficile.

Des solutions possibles pour un environnement réel seraient de détecter les infrarouges pour encore plus d'informations ou d'utiliser du machine learning.

Tracking

Une fois le traitement d'image terminé, nous utilisons les fonctions de opencv qui nous permettent de récupérer un objet rectangle autour de chaque blobs. Ensuite, si le rectangle est une nouvelle lampe détectée, nous lui créons un tracker d'opencv. Ainsi, on peut suivre les mouvements d'une lampe tout au long des frames jusqu'à ce qu'elle disparaisse.

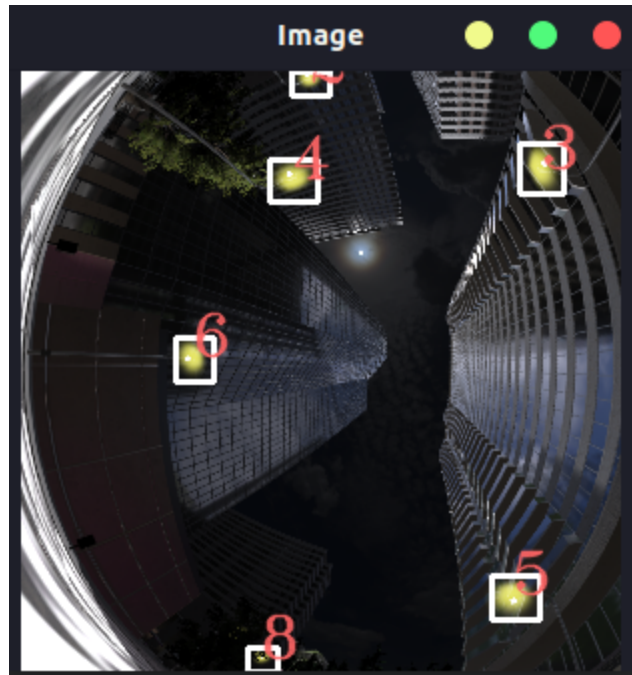


Matching et calcul de pose

Pour ce programme la partie la plus importante est de bien coupler les points vu par la caméra aux points du monde virtuel. Pour comprendre la méthode de couplage il faut déjà établir certaines choses. Tout d'abord, nous considérons que la pose initiale est connue. Nous couplons aussi de base les lampes détectées sur la première frame à leurs points 3D dans la carte virtuelle. Ainsi nous avons un environnement sain et correct au départ. Ensuite, à chaque frame nous mettons à jour les coordonnées 2D des points 3D monde avec la pose calculée précédemment. Le calcul de pose se fait avec l'algorithme virtual vs avec un ransac dedans qui calcule la pose à partir de points. C'est avec les coordonnées 2D monde et les coordonnées 2D détectées sur la frame courante que nous faisons les couplages. Pour cela nous lions les points 2D détectés avec le points 2D monde le plus proche.

Cette technique quoique simple et efficace présente des défauts. Il y a des cas où un point est lié au mauvais point monde. Si 2 points sont en compétition sur le même point et que le bon point est plus proche, le problème peut être réglé facilement. Cependant, le cas où le mauvais point est vu plus proche existe aussi ce qui complique beaucoup la tâche. En effet à cause de cela on peut tracker un mauvais point longtemps et tout fausser.

Dans cet exemple on a un tournant ce qui met (dans cet angle) deux lampes très proches, la 8 et la 10. A ce moment-là, nous avons aussi un arbre qui cache la lampe 8 mais pas la 10. On peut donc observer que l'on détecte la lampe 8 au lieu de la 10 (cf schéma de la carte plus haut).

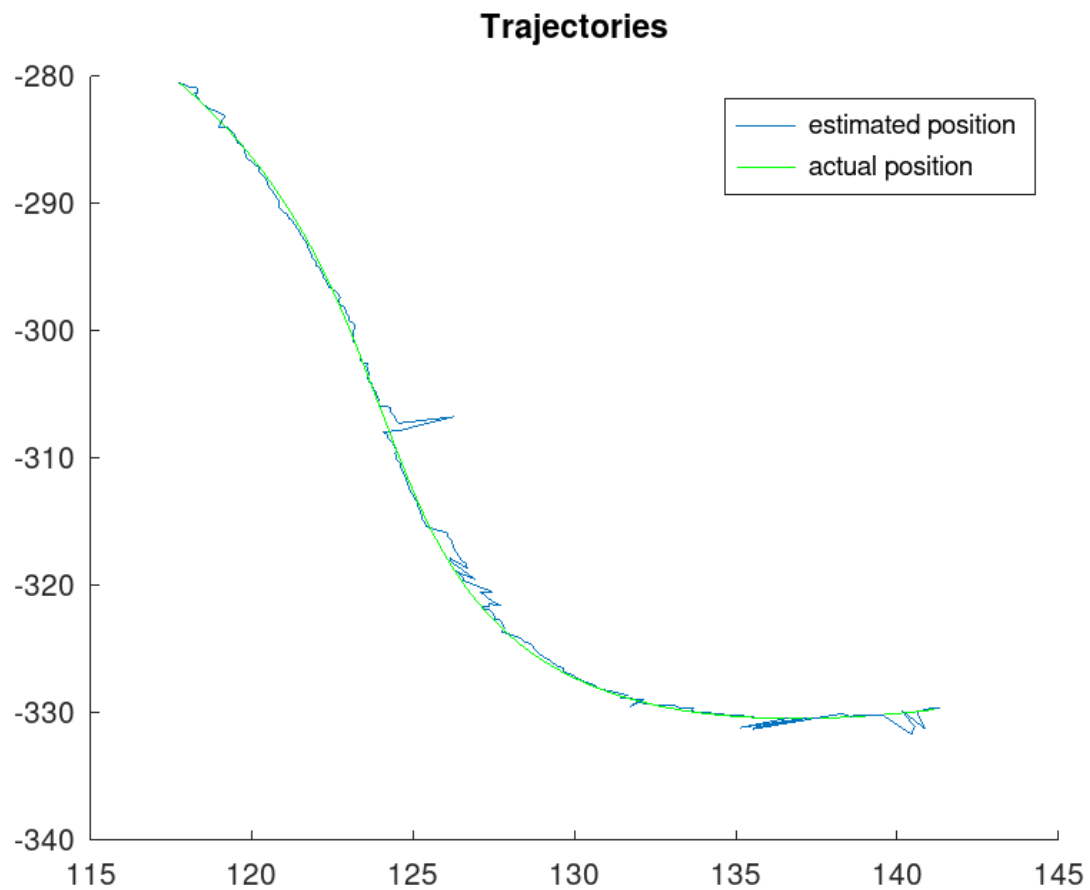


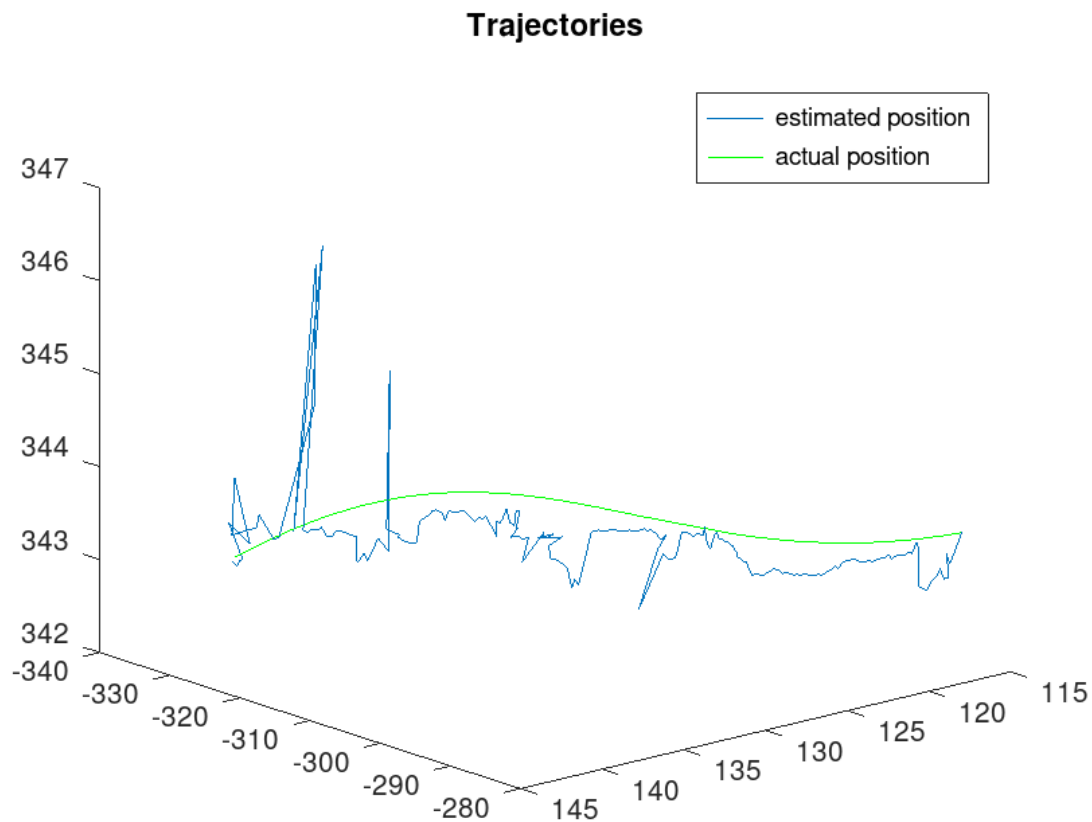
Pour résoudre ce problème, nous avons appliqué une politique assez dure pour permettre de l'auto correction. Par exemple, un point déjà tracker peut être remplacé par un nouveau point si celui-ci est plus proche. Ou encore, si 2 nouveaux points sont en concurrence pour le même point alors on ne prend ni l'un ni l'autre.

Si cette politique assez sévère permet de l'auto correction, elle présente aussi des problèmes. En supprimant trop de lampes l'estimation de pose devient moins précise et peut ne pas marcher par manque d'information.

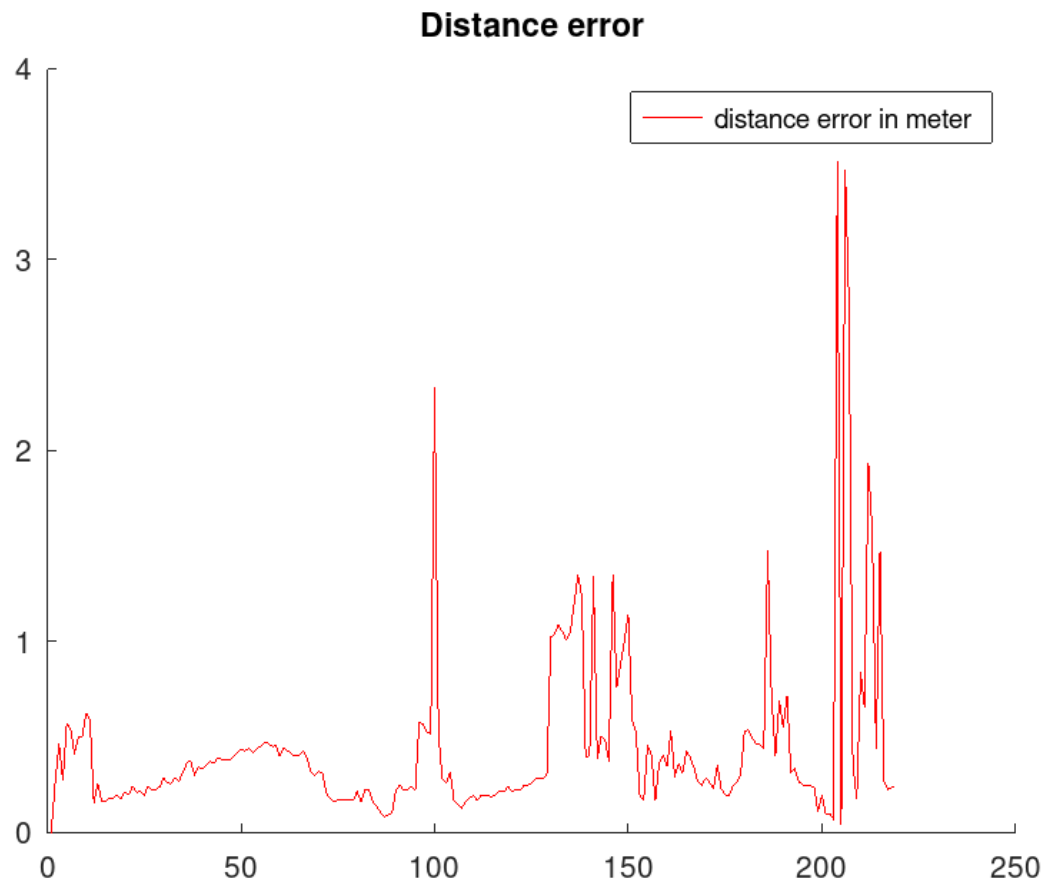
Etude et résultats

Voici des graphiques de la trajectoire obtenue par rapport à la trajectoire réelle.

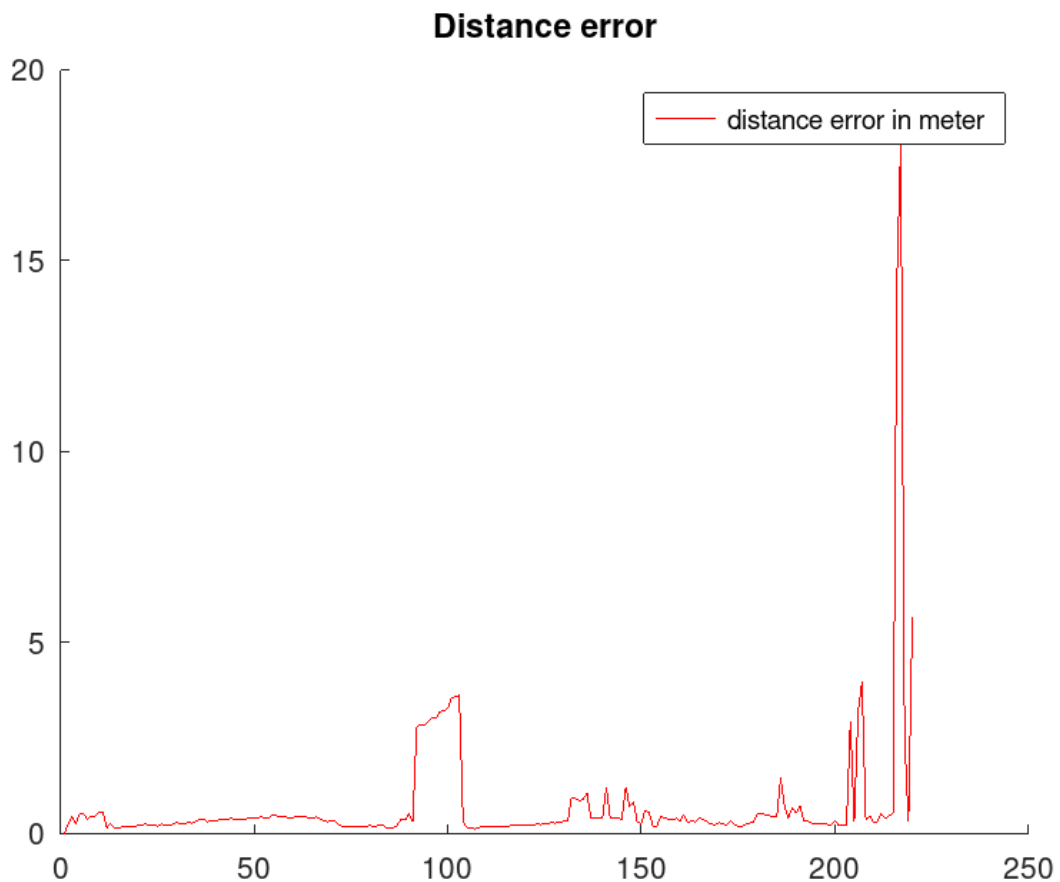




On peut observer dans ces graphiques que l'estimation suit bien la trajectoire malgré des décalages. Ce qui n'est pas encore très bien expliqué c'est que là où la pose dévie le plus c'est sur l'axe z soit la hauteur. Celle-ci, normalement constante dans notre simulation, varie beaucoup. Une hypothèse serait que de prendre le point milieu des blobs n'est pas une assez bonne représentation de la localisation de la lampe.



Notre erreur ici est plutôt encourageante même si pas à la hauteur pour des applications de véhicules autonomes. Nous restons beaucoup dans les moins de 1 mètre de distance. La politique assez sévère de matching démontre aussi le potentiel d'auto correction. Pour un contraste, voici un graphique avec une politique plus lax.



Il est important de regarder le pic autour de 100. L'exemple présenté plus tôt dans la partie matching se passe là. On dévie beaucoup à ce moment parce qu'une lampe est mal couplée. Avec la politique sévère ce pic est drastiquement réduit en supprimant cette concurrence entre les 2 lampes.

Conclusion et ouverture

Pour conclure cette approche présente de nombreux avantages et a obtenue des résultats très encourageants. Il est important de noter que ces résultats sont très préliminaires et que cette méthode n'a pas encore été poussée jusqu'au bout. Elle présente un bon avenir par son anticipation de la lifi et ces avantages considérables en termes d'optimisation. Des peaufinages et ajouts de certaines méthodes

augmenteraient drastiquement les résultats. Une méthode à ajouter pourrait être le machine learning. Cela permettra de parfaitement détecter les lampes et les coordonnées 2D.

Références

- [1] Y. KUSUNO, J. TAKAHASHI and Y. YONG, "A Method Localizing an Omnidirectional Image in Pre-Constructed 3D Wireframe Map," 2019 IEEE/SICE International Symposium on System Integration (SII), 2019, pp. 325-329, doi: 10.1109/SII.2019.8700393.
- [2] T. Cham, A. Ciptadi, W. Tan, M. Pham and L. Chia, "Estimating camera pose from a single urban ground-view omnidirectional image and a 2D building outline map," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2010, pp. 366-373, doi: 10.1109/CVPR.2010.5540191.
- [3] Xilin Ke et al, 3D Scene Localization and Mapping Based on Omnidirectional SLAM, 2021 IOP Conf. Ser.: Earth Environ. Sci. 783 012143
- [4] K. Khoshelham and M. Ramezani, "Vehicle positioning in the absence of GNSS signals: Potential of visual-inertial odometry," 2017 Joint Urban Remote Sensing Event (JURSE), 2017, pp. 1-4, doi: 10.1109/JURSE.2017.7924574.
- [5] N. Crombez, Contributions aux asservissements visuels denses : nouvelle modélisation des images adaptée aux environnements virtuels et réels, 2015.