



HAL
open science

Modélisation adaptée des images omnidirectionnelles pour agrandir le domaine de convergence de l'asservissement visuel virtuel direct

Seif Eddine Guerbas

► **To cite this version:**

Seif Eddine Guerbas. Modélisation adaptée des images omnidirectionnelles pour agrandir le domaine de convergence de l'asservissement visuel virtuel direct. Robotique [cs.RO]. Université de Picardie Jules Verne, 2022. Français. NNT : 2022AMIE0026 . tel-04057304

HAL Id: tel-04057304

<https://theses.hal.science/tel-04057304v1>

Submitted on 4 Apr 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Thèse de Doctorat

Mention Sciences pour l'Ingénieur
Spécialité Robotique

présentée à l'École Doctorale en Sciences, Technologie, Santé (ED 585)

à l'Université de Picardie Jules Verne

par

Seif Eddine GUERBAS

pour obtenir le grade de Docteur de l'Université de Picardie Jules Verne

**Modélisation adaptée des images omnidirectionnelles pour
agrandir le domaine de convergence de l'asservissement visuel
virtuel direct**

Soutenue le 13 juillet 2022 après avis des rapporteurs, devant le jury d'examen :

| | |
|---|---------------------------|
| M. C. DEMONCEAUX, PU, Université Bourgogne Franche-Comté | Président (Rapporteur) |
| M. O. STRAUSS, Mdc HDR, Université de Montpellier | Rapporteur |
| M^{me}. M. BABEL, Mdc HDR, INSA de Rennes | Examinatrice |
| M^{me}. S. AINOUZ, PU, INSA de ROUEN | Examinatrice |
| M. E. MOUADDIB, PU, Université de Picardie Jules Verne | Directeur de thèse |
| M. G. CARON, Mdc HDR, Université de Picardie Jules Verne | Co-encadrant |

Remerciements

Puisque l'écriture scientifique exige de mettre de côté les sentiments personnels du chercheur, il me reste ces quelques lignes pour exprimer ma gratitude envers celles et ceux sans qui je n'aurais pu mener à bien ce travail. Je tiens d'abord à remercier le jury d'avoir accepté d'assister à ma soutenance de thèse. Je remercie plus particulièrement Monsieur Cédric Demonceaux d'avoir présidé le jury et d'avoir accepté d'être rapporteur ainsi que Madame Samia Ainouz et Madame Marie Babel d'avoir examiné mon travail. Je voudrais exprimer ma gratitude à Monsieur Olivier Strauss qui, en tant que rapporteur, a fait un long voyage pour assister à ma soutenance et qui m'a accompagné et encadré durant mon master à l'université de Montpellier.

Je tiens également à remercier mes collègues du laboratoire MIS qui m'ont accueilli au cours de cette thèse. Je mentionne particulièrement Monsieur Gilles Dequen pour ses orientations et ses conseils, et, véritable rayon de soleil à Amiens, Madame Juliette Dubois, dont les mots ne sauraient traduire la beauté intérieure et l'humanité. Je remercie aussi Monsieur El Mustapha Mouaddib et Monsieur Guillaume Caron qui ont dirigé cette thèse.

J'exprime mon infinie gratitude et mon immense respect au soleil et à la lune de ma vie - ma mère et mon père - sans qui je n'aurais jamais obtenu mon doctorat. Je remercie mes sœurs (Sihem, Khaoula, Dr. Imene et Chaima), ainsi que ma source d'énergie, Madame Amel Fekrane, dont la bienveillance est sans limites. Je remercie Madame Anne Boffy qui m'a accueilli chaleureusement durant le MBA et qui m'a beaucoup soutenu. Merci aux professeurs que j'ai rencontrés au cours de mon parcours scolaire et qui m'ont donné goût aux sciences : je pense à ma professeure du primaire, Mme Laila, à mes professeurs du lycée, Mme Abdsmad et M. Khireddine, à mes professeurs de l'université, M. Charef, M. Bennia et M. Girard.

Enfin, je remercie infiniment Hugo Freby pour son soutien et ses corrections, Nathan Sanchez pour son aide très précieuse, les jeunes docteurs et doctorant du MIS (Nathan Crombez, Jordan Caracotte et Louis Couturier) et mes amis qui m'ont accompagné durant ces années (Rokia Aouati, Housseem-Eddine Benseddik, Pierre Moreau, Vivien Canthaleu, Loïc Bosco, Guéric Valantin, Audrey Ambles, Sébastien Cohen, Jérémie Landrieu).

J'aimerais dédicacer cette thèse à ma grand-mère Aïcha GUERBAS que j'ai tant aimée et qui est décédée quelques mois avant l'achèvement de ce long travail.

Table des matières

| | |
|---|-------------|
| Remerciements | i |
| Table des sigles et acronymes | xiii |
| Préambule | 1 |
| Contributions | 3 |
| 1 Asservissement visuel photométrique omnidirectionnel : modélisation et état de l’art ciblé | 5 |
| 1.1 Introduction | 6 |
| 1.2 Modélisation géométrique de caméra | 7 |
| 1.2.1 Vision perspective | 7 |
| 1.2.2 Vision omnidirectionnelle | 9 |
| 1.2.3 De l’objet à la caméra | 15 |
| 1.3 Différentes approches de l’asservissement visuel | 16 |
| 1.3.1 Asservissement visuel basé primitives géométriques | 17 |
| 1.3.2 Asservissement visuel photométrique | 19 |
| 1.4 Application de l’asservissement visuel photométrique omnidirectionnel dans un environnement virtuel | 23 |
| 1.4.1 Protocole | 24 |
| 1.4.2 Résultats | 25 |
| 1.5 Conclusion | 34 |
| 2 Asservissement visuel omnidirectionnel basé mélanges de Gaussiennes photométriques | 35 |
| 2.1 Introduction | 36 |

| | | |
|----------|--|-----------|
| 2.2 | Modélisation d'une image optique | 36 |
| 2.3 | Représentation abstraite des pixels d'une image | 37 |
| 2.3.1 | Modèle de gaussienne photométrique | 37 |
| 2.3.2 | Mélange de gaussiennes photométriques | 40 |
| 2.4 | Mélanges de gaussiennes photométriques comme caractéristiques visuelles denses en asservissement visuel omnidirectionnel | 41 |
| 2.4.1 | Loi de commande | 42 |
| 2.4.2 | Impact du paramètre d'extension | 44 |
| 2.4.3 | Règle existante pour choisir les paramètres d'extension | 45 |
| 2.4.4 | Nouvelles règles pour choisir les paramètres d'extension | 46 |
| 2.5 | Évolution en simulation | 46 |
| 2.5.1 | Protocole | 46 |
| 2.5.2 | Ordres des grandeurs des MGP selon les 3 règles | 47 |
| 2.5.3 | Simulation en 1 DDL | 48 |
| 2.5.4 | Simulation en 2 DDL (2 translations) | 53 |
| 2.5.5 | Simulation en 3 DDL (3 translations) | 56 |
| 2.5.6 | Simulations en 6 DDL (3 translations et 3 rotations) | 59 |
| 2.5.7 | Évaluation multiple du domaine de convergence | 65 |
| 2.6 | Conclusion | 66 |
| 3 | Suivi 3D par asservissements visuels virtuels successifs | 69 |
| 3.1 | Introduction | 70 |
| 3.2 | Alignement dense direct d'images omnidirectionnelles sur nuage de points 3D | 71 |
| 3.3 | Suivi 3D par alignements successifs | 72 |
| 3.4 | Résultats expérimentaux | 73 |
| 3.4.1 | Obtention des nuages de points 3D source | 73 |
| 3.4.2 | Évaluation de la robustesse à la pose initiale | 73 |

| | | |
|----------|---|-----------|
| 3.4.3 | Application sur de longues trajectoires | 77 |
| 3.5 | Conclusion | 81 |
| 4 | Conclusion et perspectives | 83 |
| | Conclusion | 83 |

Table des figures

| | | |
|------|---|----|
| 1.1 | Principe de l’asservissement visuel (images extraites de [COLLEWET et al., 2008a], figure du robot extraite de [SILVEIRA et al., 2020]). | 6 |
| 1.2 | Principe du fonctionnement de la caméra selon Ibn Al-Haytham [AL-HASSANI, 2007] | 8 |
| 1.3 | Modèle de projection perspective de la scène vers le plan pixellique | 10 |
| 1.4 | Lentille <i>Fisheye</i> : (a) Lentille FUJINON <i>Fisheye</i> 1 :1.4/1.8mm FE185C057HA-1, (b) Exemple d’image panoramique [CARON et al., 2018] | 10 |
| 1.5 | Caméra perspective montée sur un système rotatif (https://www.ixbt.com/) | 11 |
| 1.6 | Système multicaméra Dodeca 2 360 (Immersive Media) | 11 |
| 1.7 | (a) Système catadioptrique, (b) Image panoramique | 12 |
| 1.8 | (a) M. C. Escher, Nature morte à la sphère réfléchissante, lithographie, 1934, (b) zoom sur la surface réfléchissante. | 13 |
| 1.9 | Modèle de projection unifié avec une sphère virtuelle [CARON, 2010]. | 14 |
| 1.10 | Transformation rigide d’un point d’objet dans le repère caméra. | 16 |
| 1.11 | Modélisation du flot optique. | 20 |
| 1.12 | Gradient : (a) image d’intensité, (b) gradient selon la direction de x, (c) gradient selon la direction de y. | 22 |
| 1.13 | (a) : Environnement réel (vue perspective), (b) : Rendu omnidirectionnel de l’environnement utilisé sur Unity pour l’asservissement visuel | 24 |
| 1.14 | Fonction de coût pour un DDL. | 26 |
| 1.15 | Simulation en 1-DDL, erreur initiale : $\Delta_X = 1.8$ m est : (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l’évolution de l’erreur et (e) l’évolution de la position de la caméra. | 27 |
| 1.16 | Fonction de coût pour deux DDL. | 27 |
| 1.17 | Simulation en 2-DDL, erreur initiale : $[\Delta_X, \Delta_Y] = [-2 \text{ m}, -1 \text{ m}]$, (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l’évolution de l’erreur et (e) l’évolution de la position de la caméra. | 28 |

| | | |
|------|---|----|
| 1.18 | Simulation en 2-DDL bis, erreur initiale : $[\Delta_X, \Delta_Y] = [-10 \text{ m}, -4 \text{ m}]$: (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l'évolution de l'erreur et (e) l'évolution de la position de la caméra. | 29 |
| 1.19 | Simulation en 3-DDL : erreur initiale : $[\Delta_X, \Delta_Y, \Delta_Z] = [1 \text{ m}, -1 \text{ m}, -1 \text{ m}]$, (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l'évolution de l'erreur et (e) l'évolution de la position de la caméra. | 30 |
| 1.20 | Simulation en 3-DDL : erreur initiale : $[\Delta_X, \Delta_Y, \Delta_Z] = [-2 \text{ m}, -1 \text{ m}, -3 \text{ m}]$, (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l'évolution de l'erreur et (e) l'évolution de la position de la caméra. | 31 |
| 1.21 | Simulation en 6-DDL : erreur initiale : $[\Delta_X, \Delta_Y, \Delta_Z, \Delta_{R_X}, \Delta_{R_Y}, \Delta_{R_Z}] = [2 \text{ m}, -2 \text{ m}, -2 \text{ m}, -5^\circ, -5^\circ, 15^\circ]$, (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l'évolution de la position et (e) l'évolution de l'orientation de la caméra. | 32 |
| 1.22 | Simulation en 6-DDL bis : erreur initiale : $[\Delta_X, \Delta_Y, \Delta_Z, \Delta_{R_X}, \Delta_{R_Y}, \Delta_{R_Z}] = [10 \text{ m}, -4 \text{ m}, -3 \text{ m}, -10^\circ, -10^\circ, 15^\circ]$, (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l'évolution de la position et (e) l'évolution de l'orientation de la caméra. | 33 |
| 1.23 | Simulation en 6-DDL ter : erreur initiale : $[\Delta_X, \Delta_Y, \Delta_Z, \Delta_{R_X}, \Delta_{R_Y}, \Delta_{R_Z}] = [-8 \text{ m}, -8 \text{ m}, -6 \text{ m}, -6^\circ, -13^\circ, -38^\circ]$, (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l'évolution de la position et (e) l'évolution de l'orientation de la caméra. | 34 |
| 2.1 | Schéma représentant le produit du signal continu (a) avec un peigne (b), donnant lieu à une image numérique (c). | 38 |
| 2.2 | Fonction gaussienne en deux dimensions centrée en $(u, v) = (0, 0)$ | 39 |
| 2.3 | Exemple d'un mélange de deux gaussiennes monodimensionnelles, calculées avec un paramètre d'extension faible (a) et avec un paramètre d'extension suffisant pour qu'il y ait chevauchement (b). | 40 |
| 2.4 | Mélange de gaussiennes photométriques appliqué sur une image d'une mire omnidirectionnelle (a) avec différentes valeurs du paramètre d'extension : $\lambda_g = 0.1$ (b), $\lambda_g = 5$ (c), $\lambda_g = 10$ (d) et $\lambda_g = 12$ (e). | 41 |
| 2.5 | Comparaison des fonctions de coût en 2-DDL : (a) image minimaliste, (b) fonction de coût basée sur la différence photométrique, (c) et (d) fonctions de coût à partir des images représentées avec le mélange de gaussiennes photométriques, pour (c) $\lambda_g^* = 7.5$ et $\lambda_g = 15$, pour (d) $\lambda_g^* = \lambda_g = 15$ | 45 |

| | | |
|------|---|----|
| 2.6 | Évaluation du domaine de convergence : (a) Visualisation d'un parallélépipède englobant les 64 positions initiales de la caméra (flèches dorées) autour d'une position souhaitée (flèche bleue).; (b-h) Images rendues aux sept poses souhaitées. | 47 |
| 2.7 | Amplitude des MGP : (a) MGP initial, (b) MGP désiré <i>Règle0</i> , (c) MGP désiré <i>Règle1</i> . | 48 |
| 2.8 | Simulation en 1-DDL règle 0 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution de la vitesse et (j) Évolution de la position de la caméra virtuelle. | 50 |
| 2.9 | Simulation en 1-DDL règle 0 bis : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution de la vitesse et (j) Évolution de la position de la caméra virtuelle. | 51 |
| 2.10 | Simulation en 1-DDL règle 1 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution de la vitesse et (j) Évolution de la position de la caméra virtuelle. | 52 |
| 2.11 | Simulation en 1-DDL règle 2 | 53 |
| 2.12 | Simulation en 2-DDL règle 0 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution des vitesses linéaires et (j) Évolution de la position de la caméra virtuelle. | 54 |
| 2.13 | Simulation en 2-DDL règle 1 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution des vitesses linéaires et (j) Évolution de la position de la caméra virtuelle. | 55 |
| 2.14 | Simulation en 2-DDL règle 2 | 56 |
| 2.15 | Simulation en 3-DDL règle 0 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution des vitesses linéaires et (j) Évolution de la position de la caméra virtuelle. | 57 |
| 2.16 | Simulation en 3-DDL règle 1 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution des vitesses linéaires et (j) Évolution de la position de la caméra virtuelle. | 58 |
| 2.17 | Simulation en 3-DDL règle 2. | 59 |
| 2.18 | Simulation en 6-DDL règle 0 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution de la position de la caméra et (j) Évolution de l'orientation de la caméra virtuelle. | 60 |
| 2.19 | Simulation en 6-DDL règle 1 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution de la position de la caméra et (j) Évolution de l'orientation de la caméra virtuelle. | 61 |

| | | |
|------|--|----|
| 2.20 | Simulation en 6-DDL règle 2 | 62 |
| 2.21 | Simulation en 6-DDL bis règle 0 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution de la position de la caméra et (j) Évolution de l'orientation de la caméra virtuelle. | 63 |
| 2.22 | Simulation en 6-DDL bis règle 1 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution de la position de la caméra et (j) Évolution de l'orientation de la caméra virtuelle. | 64 |
| 2.23 | Simulation en 6-DDL bis règle 2 : (d) Évolution de la position de la caméra et (e) Évolution de l'orientation de la caméra virtuelle. | 65 |
| 2.24 | Distributions cumulées des erreurs de positions finales | 66 |
| 3.1 | (a) : Image courante issue du modèle 3D de la ville d'Amiens " <i>Rue Victor Hugo</i> ", (b) : Image désirée issue d'une caméra réelle dans la même rue. | 75 |
| 3.2 | Suivi direct basé sur un modèle 3D avec le SoMGP et le SoP en utilisant 1 image toutes les N images(s) : (a-b) Trajectoires estimées sur (c) un sol irrégulier et difficile où des images comme (d) sont acquises ; (e-f) Superposition des dernières images acquises et des images virtuelles optimales ($N = 25$) : (e) SoP, (f) SoMGP. | 76 |
| 3.3 | Suivi direct basé sur un modèle 3D avec SoMGP et SoP avec la caméra à bord d'un véhicule. Trajectoires estimées pour (a) la première séquence S1 (GPS : bleu ; poses de la caméra issue du processus de photogrammétrie : vert foncé ; SoMGP : orange ; SoP : jaune) et (b) la deuxième séquence S2 (GPS : rouge ; SoMGP : rose ; SoP : jaune). Les trajectoires sont tracées après conversion en coordonnées de latitude et de longitude exportées sous forme de fichiers kml vers Google My Maps (carte de base monochrome). | 78 |
| 3.4 | Exemples d'images successives de la séquence S2 avec (a-b) un mouvement inter-image moyen et (c-d) un mouvement inter-image important. Les images (a-b) sont acquises au début de la trajectoire (où le SoP échoue dans la figure 3.3 - b) et (c-d) dans le virage. | 79 |
| 3.5 | Superposition des dernières images virtuelles acquises et optimales ($N = 1$) : (a) et (c) correspondent respectivement au début des trajectoires rose et jaune de la figure 3.3 - b, alors que (b) et (d) correspondent à la fin de ces trajectoires. | 80 |

Liste des tableaux

| | | |
|-----|---|----|
| 3.1 | Les avantages et les limites de la numérisation par photogrammétrie et par scanner laser [HÉNO et al., 2014]. | 74 |
|-----|---|----|

Table des sigles et acronymes

| | |
|--------------|--|
| DDL | <i>Degré De Liberté</i> |
| AV | <i>Asservissement Visuel</i> |
| MGP | <i>Mélange de Gaussiennes Photométriques</i> |
| AVPo | <i>Asservissement Visuel Photométrie omnidirectionnel</i> |
| SoMGP | <i>Suivi omnidirectionnel basé Mélange de Gaussiennes Photométriques</i> |
| SoP | <i>Suivi omnidirectionnel Photométrie</i> |
| SSD | <i>Sum of Squared Difference</i> |
| NCC | <i>Normalized Cross Correlation</i> |
| ZSSD | <i>Zero-mean Sum of Squared Differences</i> |
| NSSD | <i>Normalized Sum of Squared Differences</i> |
| ZNSSD | <i>Zero-mean Normalized Sum of Squared Differences</i> |

Préambule

En robotique, les capteurs visuels sont largement utilisés afin d'acquérir des informations riches sur l'environnement. Les caméras perspectives utilisées traditionnellement offrent toutefois une perception limitée de celui-ci. Au contraire, la caméra omnidirectionnelle, qui peut résulter de la combinaison d'une caméra conventionnelle et d'un miroir courbé, permet d'élargir son champ de vue qui peut atteindre 180° et plus.

L'axe de recherche de cette thèse est l'exploitation des avantages des caméras omnidirectionnelles pour un suivi visuel 3D de modèle d'environnement permettant la localisation du robot ou du véhicule qui l'embarque. Ce suivi 3D repose sur le concept d'asservissement visuel virtuel en vision omnidirectionnelle. L'asservissement visuel omnidirectionnel consiste à contrôler en boucle fermée la position et l'orientation d'un système dynamique dans l'espace de travail en utilisant les données visuelles comme retour d'information. Afin d'établir la relation entre les images omnidirectionnelles provenant d'une caméra réelle ou virtuelle et ses déplacements dans l'environnement, deux approches peuvent être utilisées. La première consiste à utiliser des primitives géométriques, comme des points et des droites, extraites de l'image désirée et de celles acquises tout au long de l'asservissement visuel. Cependant, ce type d'approche nécessite une phase de traitement d'images et la précision de l'asservissement visuel dépend dans une large mesure de la précision de l'extraction de ces primitives géométriques. La seconde consiste à utiliser toute l'information photométrique présente dans les images afin d'éliminer la détection et le suivi de primitives géométriques.

C'est donc sur ce second type d'approche qu'est fondé notre propos, que nous proposons d'articuler en quatre temps. Les différentes modélisations géométriques de la caméra perspective et omnidirectionnelle sont d'abord abordées ainsi que les différentes approches de l'asservissement visuel, qu'il soit basé primitives géométrique ou information purement photométrique. L'asservissement visuel photométrique omnidirectionnel peut ensuite être appliqué dans un environnement virtuel 3D constitué à partir d'un nuage de points obtenu, par exemple, par un scanner-laser. Les résultats mettent en lumière les limites de l'asservissement visuel photométrique.

C'est pourquoi le deuxième chapitre se propose d'élargir le domaine de convergence. La pose optimale de la caméra peut répondre au problème d'optimisation non linéaire. La convergence du capteur visuel de la pose initiale vers la pose désirée dépend dans une large mesure de la distance qui sépare les deux. Or, en adaptant une représentation d'image basée mélange de gaussiennes photométriques aux images omnidirectionnelles et en améliorant la règle de paramétrisation, le domaine de convergence est élargi de manière considérable. Différentes expérimentations d'asservissement visuel basé mélange de gaussiennes photométriques, menées avec différentes règles sur des images omnidirectionnelles courantes et désirées obtenues avec la même caméra virtuelle, le démontrent.

Le troisième chapitre applique l'asservissement visuel virtuel omnidirectionnel basé mélange de gaussiennes photométriques avec les nouvelles règles sur les paramètres à l'estimation de pose relative de la caméra réelle par rapport à la scène en se basant sur la connaissance du modèle 3D de la scène réelle. Les expérimentations sur des séquences d'images acquises à l'extérieur en ville sur robot mobile et sur véhicule montrent clairement l'apport avec une grande robustesse aux mouvements inter-image importants permettant un suivi 3D beaucoup plus long que les méthodes concurrentes de l'état de l'art.

Enfin, la conclusion résume les contributions de la thèse et propose, à partir de celles-ci, quelques perspectives de recherche.

Contributions

L'ensemble de ces nouvelles contributions est développé dans les trois articles suivants que nous avons publiés pendant la préparation de la thèse :

- [S. E. GUERBAS et al., 2021] S.E. Guerbas, N. Crombez, G. Caron et E.M. Mouaddib. Direct 3D model-based tracking in omnidirectional images robust to large inter-frame motion, In *2021 20th International Conference on Advanced Robotics (ICAR)*.
- [S.-E. GUERBAS et al., 2021] S.E. Guerbas, N. Crombez, G. Caron et E.M. Mouaddib. Photometric Gaussian Mixtures for Direct Virtual Visual Servoing of Omnidirectional Camera, In *IEEE CVPR 2021 Workshop on 3D Vision and Robotics*, Seattle (virtual), United States.
- [GUERBAS et al., 2021] S.E. Guerbas, N. Crombez, G. Caron et E.M. Mouaddib. Mélange de Gaussiennes Photométriques pour l'Asservissement Visuel Virtuel Direct d'une Caméra Omnidirectionnelle, In *ORASIS 2021*, Saint Ferréol, France. Centre National de la Recherche Scientifique [CNRS].

Asservissement visuel photométrique omnidirectionnel : modélisation et état de l'art ciblé

Sommaire

| | | |
|------------|--|-----------|
| 1.1 | Introduction | 6 |
| 1.2 | Modélisation géométrique de caméra | 7 |
| 1.2.1 | Vision perspective | 7 |
| 1.2.2 | Vision omnidirectionnelle | 9 |
| 1.2.3 | De l'objet à la caméra | 15 |
| 1.3 | Différentes approches de l'asservissement visuel | 16 |
| 1.3.1 | Asservissement visuel basé primitives géométriques | 17 |
| 1.3.2 | Asservissement visuel photométrique | 19 |
| 1.4 | Application de l'asservissement visuel photométrique omnidirectionnel dans un environnement virtuel | 23 |
| 1.4.1 | Protocole | 24 |
| 1.4.2 | Résultats | 25 |
| 1.5 | Conclusion | 34 |

1.1 Introduction

L'asservissement visuel d'un robot permet le contrôle en boucle fermée dans l'espace de travail de la position et l'orientation de la caméra [AHLUWALIA et al., 1986]. Ce processus a d'abord été appliqué à des bras robotisés industriels dans les années 70 [BOLLES et al., 1973], avant d'être étendu aux robots mobiles [CONTICELLI et al., 1999]. La commande en boucle fermée est construite à partir des données visuelles capturées dans la scène.

Il existe plusieurs configurations :

- **La configuration dite *eye-to-hand*** : la caméra peut être ancrée sur une plate-forme stationnaire comme sur un autre dispositif mécanique [JARVIS, 1977] et être disposée de manière à pouvoir observer le robot, son organe terminal [FLANDIN et al., 2000], sa configuration articulaire [RUF et al., 2000] et les différents objets qui se trouvent dans son espace de travail [HORAUD et al., 1998].
- **La configuration dite *eye-in-hand*** : Il s'agit de la configuration la plus répandue. Dans ce type de configuration, la caméra est fixée de manière rigide sur le robot (à l'organe terminal pour les robots manipulateurs) afin d'observer les différents objets qui se trouvent dans l'espace de travail (Fig. 1.1). La plupart des travaux cités dans le reste du document emploie cette configuration.

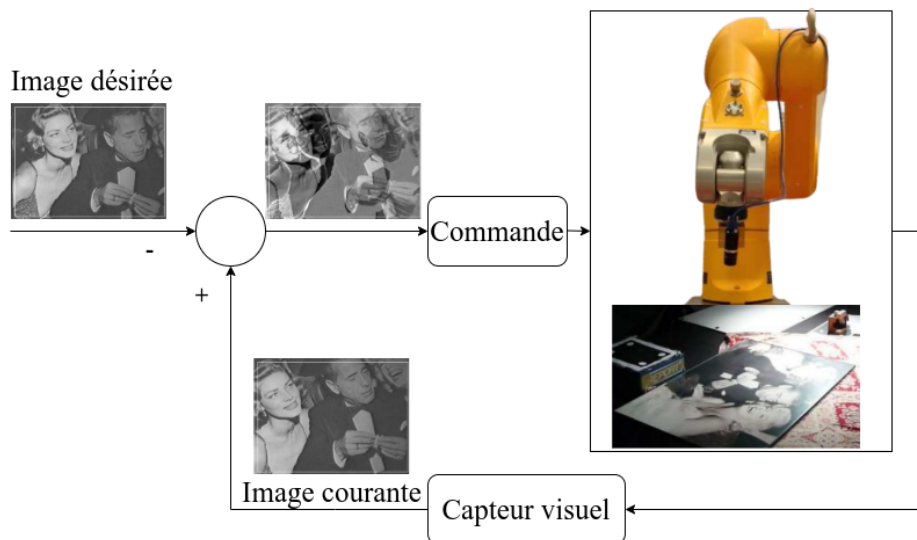


FIGURE 1.1 – Principe de l'asservissement visuel (images extraites de [COLLEWET et al., 2008a], figure du robot extraite de [SILVEIRA et al., 2020]).

La loi de commande permet de déplacer la caméra d'une position initiale vers une position désirée. Le mouvement s'arrête lorsque l'image courante acquise par la caméra et l'image acquise antérieurement à la position désirée sont identiques, à une tolérance près.

Nous nous proposons dans le cadre de ce chapitre de présenter d'abord les modélisations

des caméras perspective et omnidirectionnelle avant de décrire la scène observée par la caméra et d'expliquer les différentes approches de l'asservissement visuel. Nous proposerons enfin d'appliquer l'une d'entre elles, basée sur la photométrie, dans un environnement virtuel pour en illustrer le comportement, les avantages et les limites.

1.2 Modélisation géométrique de caméra

Le terme "caméra" est issu du latin "camera obscura", qui signifie "espace sombre fermé", une traduction de l'arabe "El Komra" signifiant "chambre noire". Ibn Al-Haytham¹ a utilisé cette "chambre noire" pour effectuer des expériences dans le domaine de la vision.

De cette manière, il a pu étudier l'optique des trous d'aiguille en plaçant des bougies en plusieurs emplacements devant l'un d'entre eux. Il a ainsi pu observer les images inversées (gauche et droite) et renversées (haut et bas) qui en résultaient sur le mur de la chambre obscure (Fig.1.2). Cette expérience a été précisément décrite dans son livre d'optique intitulé en arabe, *Kitab Al-Manazir* [A. M. SMITH, 2001], ouvrage dans lequel a été établie la démonstration de la linéarité de la lumière [J. D. SMITH, 1992].

1.2.1 Vision perspective

Le modèle de projection le plus utilisé en vision par ordinateur est toujours celui proposé par Ibn Al-Haytham. Une chambre avec un trou minuscule sur l'un de ses murs appelé "trou d'aiguille", est le point de convergence des faisceaux lumineux provenant de la scène. La quantité de ces derniers contenue dans un volume pyramidal dépend, dans une large mesure, de la taille de l'image tapissant le mur de la chambre opposé au trou d'aiguille. Un autre paramètre, primordial, est à prendre en considération : la distance entre le trou d'aiguille et l'image, à savoir la focale. Idéalement, chaque point 3D de la scène est représenté par un rayon lumineux qui va traverser le trou pour se projeter sur le mur au fond de la chambre. L'ensemble de ces faisceaux lumineux forme une image inversée et renversée en deux dimensions de la scène qui elle, est en trois dimensions.

Les caméras que nous utilisons aujourd'hui s'inspirent de la chambre noire décrite ci-dessus. Le trou de convergence des rayons lumineux est appelé "centre optique" et le mur de la chambre noire correspond à la matrice de cellules photosensibles des caméras numériques.

1. Abu Ali Al-Hasan Ibn Al-Hasan Ibn Al-Haytham (Bassora, 965 – Le Caire, 1039) est un mathématicien, philosophe, physiologiste et physicien du monde médiéval arabo-musulman, un des premiers promoteurs de la méthode scientifique expérimentale, mais aussi un des premiers physiciens théoriciens à utiliser les mathématiques. Il s'illustre par ses travaux fondateurs dans le domaine de l'optique physiologique et géométrique. C'est pour ces raisons qu'il est décrit comme le premier véritable scientifique, héritier des savants grecs et indiens [HAÏDAR, 2021].

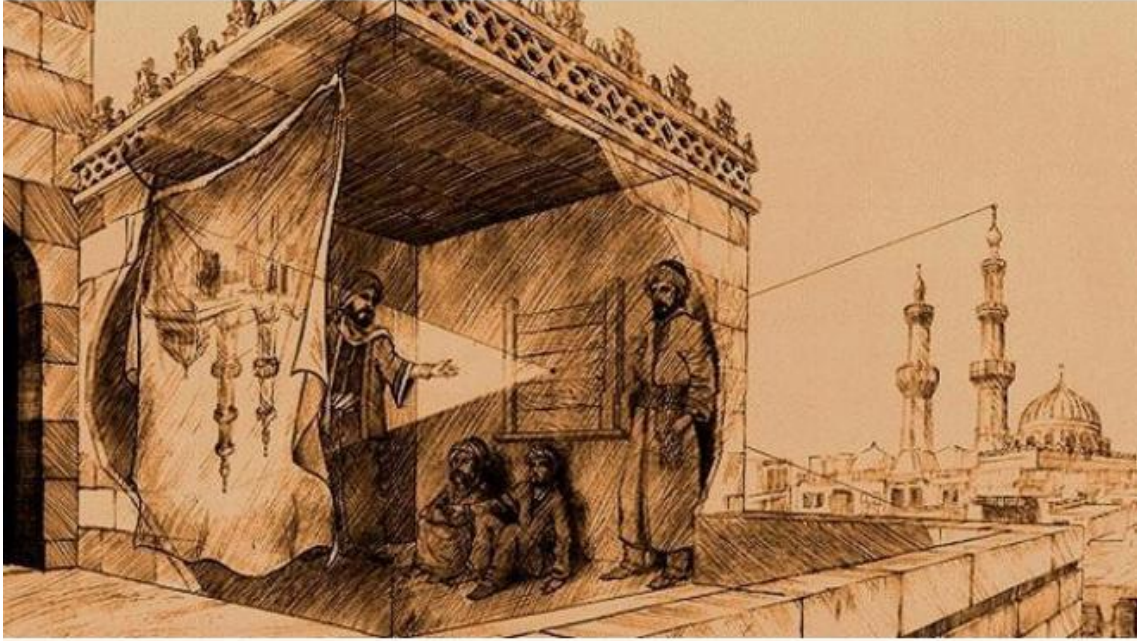


FIGURE 1.2 – Principe du fonctionnement de la caméra selon Ibn Al-Haytham [AL-HASSANI, 2007]

1.2.1.1 Projection perspective

Nous représentons plus formellement la formation géométrique de l'image par la fonction de projection perspective $Pr()$ d'un point 3D de la scène ${}^c\mathbf{X} = ({}^cX, {}^cY, {}^cZ, 1)^T$ au point 2D de l'image $\mathbf{x} = (x, y, 1)^T$:

$$\mathbf{x} = Pr({}^c\mathbf{X}) \text{ avec, } \begin{cases} x = \frac{{}^cX}{{}^cZ} \\ y = \frac{{}^cY}{{}^cZ} \end{cases} \quad (1.1)$$

${}^c\mathbf{X}$ est exprimé dans le repère caméra ${}^c\mathcal{F}$, le point résultant de la projection perspective \mathbf{x} est exprimé dans le plan image normalisé ${}^i\mathcal{F}$.

1.2.1.2 Passage au plan image numérique

Après la projection du point 3D ${}^c\mathbf{X}$ sur le plan image normalisé, un changement d'échelle est nécessaire pour exprimer le point \mathbf{x} du plan image normalisé vers une représentation d'image pixellique de repère ${}^p\mathcal{F}$. L'origine de ce repère se situe dans le coin supérieur gauche alors que celle de ${}^i\mathcal{F}$ se trouve au centre du plan image. Les changements d'échelle et de repère nécessaires pour passer du repère d'image normalisé ${}^i\mathcal{F}$ au repère d'image pixellique ${}^p\mathcal{F}$ dépendent des paramètres intrinsèques de la caméra : $\eta = \{\alpha_u, \alpha_v, u_0, v_0\}$. La transformation

réalisée est donnée par :

$$\mathbf{u} = \mathbf{K}\mathbf{x}, \quad (1.2)$$

avec $\mathbf{u} = (u, v, 1)^T$ l'emplacement du point \mathbf{x} dans le repère pixellique et la matrice \mathbf{K} qui contient les paramètres intrinsèques de la caméra :

$$\mathbf{K} = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (1.3)$$

Les paramètres du changement d'échelle α_u, α_v sont directement liés à la focale f et aux paramètres k_u, k_v , respectivement la largeur et la hauteur du pixel sur la cellule photosensible. Enfin, u_0 et v_0 sont les coordonnées du point principal dans l'image.

La combinaison de l'équation 1.1 avec la matrice intrinsèque \mathbf{K} conduit au modèle de projection complet d'un point 3D ${}^c\mathbf{X}$, exprimé dans le repère ${}^c\mathcal{F}$, vers un emplacement pixellique représenté dans le repère ${}^p\mathcal{F}$:

$$\mathbf{u} = Pr_\eta({}^c\mathbf{X}) = \mathbf{K}Pr({}^c\mathbf{X}) \quad (1.4)$$

La Figure 1.3 représente le schéma complet de la projection du point 3D ${}^c\mathbf{X}$ dans la scène, vers le plan pixellique.

La caméra perspective est largement utilisée en robotique et en vision par ordinateur mais souffre d'un champ de vue restreint. En effet, le capteur de la caméra perspective ne perçoit qu'un volume pyramidal de l'ensemble des faisceaux lumineux, ce qui borne le champ de vue. Un champ plus large, panoramique voire même complètement sphérique, est souvent nécessaire en robotique mobile [NAYAR, 1997a].

1.2.2 Vision omnidirectionnelle

Les caméras omnidirectionnelles captent des images panoramiques, voire sphériques. Trois approches permettent d'acquérir des images omnidirectionnelles.

La première consiste à utiliser des lentilles spécifiques (dioptries) pour élargir d'une façon considérable le champ de vue d'une caméra conventionnelle [MIYAMOTO, 1964]. Elle permet d'obtenir une vision panoramique grâce aux lentilles panoramiques [NAYAR, 1997b], et «*fish-eye*» [XIONG et al., 1997], jusqu'à 280° de champ de vue avec l'objectif *Fisheye d'Entaniya M12 280*. La figure 1.4 - a montre un exemple d'objectif *fish-eye* et l'image obtenue par une caméra l'utilisant est donnée en figure 1.4 - b.

La deuxième approche consiste à générer une image mosaïque combinant une série d'images

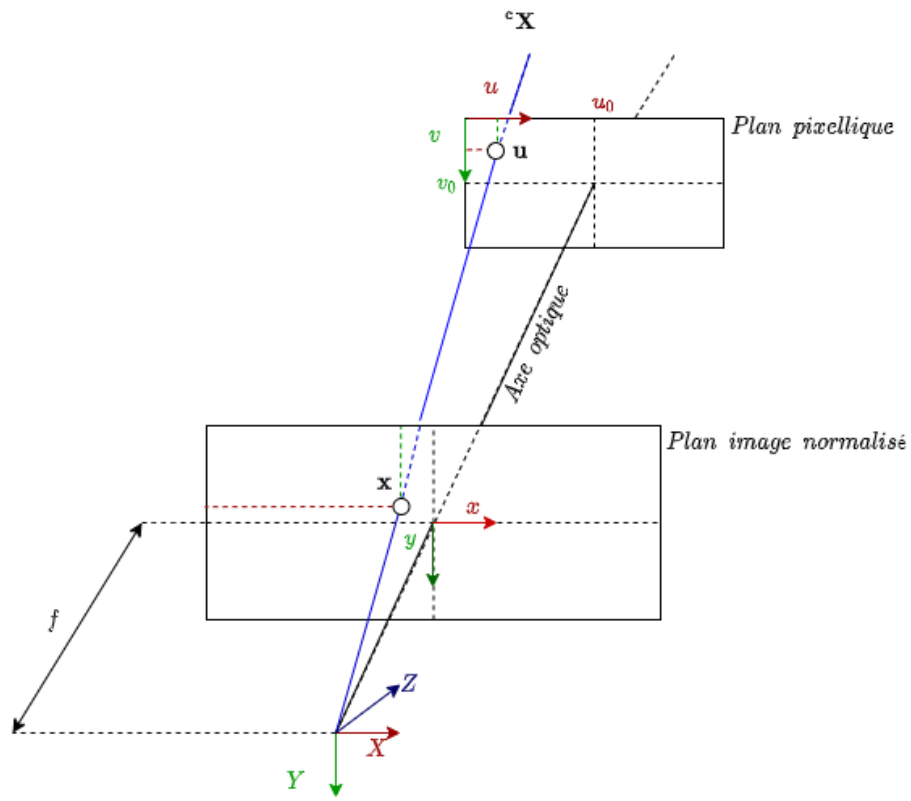


FIGURE 1.3 – Modèle de projection perspective de la scène vers le plan pixellique

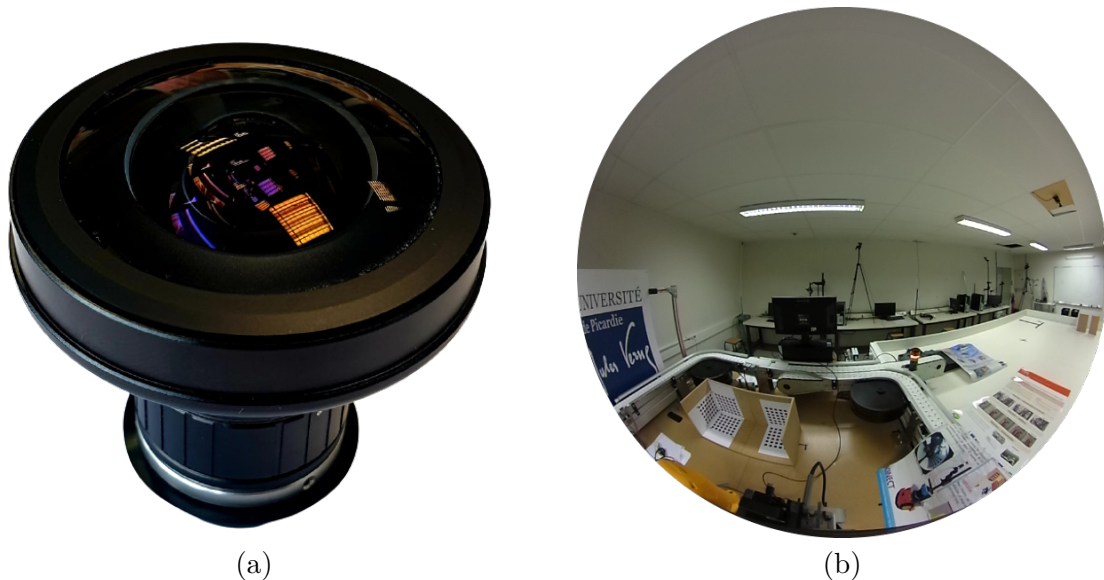


FIGURE 1.4 – Lentille *Fisheye* : (a) Lentille FUJINON *Fisheye* 1 :1.4/1.8mm FE185C057HA-1, (b) Exemple d'image panoramique [CARON et al., 2018]

distinctes, obtenues avec une caméra qu'on déplace [DANSEREAU et al., 2017], ou bien avec plusieurs caméras conventionnelles [COSSAIRT et al., 2011]. Cette technique a l'avantage de générer une image de haute résolution. Ainsi, plusieurs configurations de caméras conventionnelles peuvent être employées pour générer la mosaïque. Par exemple, il est possible de faire pivoter une caméra perpendiculairement à son axe optique (Figure 1.5). La complexité d'une telle méthode réside dans la mise en correspondance des différentes images pour construire la mosaïque [LAI et al., 2011]. Il est également possible d'installer plusieurs caméras conventionnelles sur une sphère (Figure 1.6), de telle sorte que tous les axes optiques associés se croisent au centre de la sphère [PIERCE et al., 2005]. Ces systèmes nécessitent de synchroniser toutes les acquisitions.



FIGURE 1.5 – Caméra perspective montée sur un système rotatif (<https://www.ixbt.com/>)



FIGURE 1.6 – Système multicaméra Dodeca 2 360 (Immersive Media)

La troisième et dernière approche consiste à combiner une caméra conventionnelle avec un miroir [REES, 1970] (Fig. 1.7 - a), afin que la caméra puisse observer la projection de la scène sur ledit miroir (Fig. 1.7 - b). Ce dernier peut être conique [PEGARD et MOUADDIB, 1996],

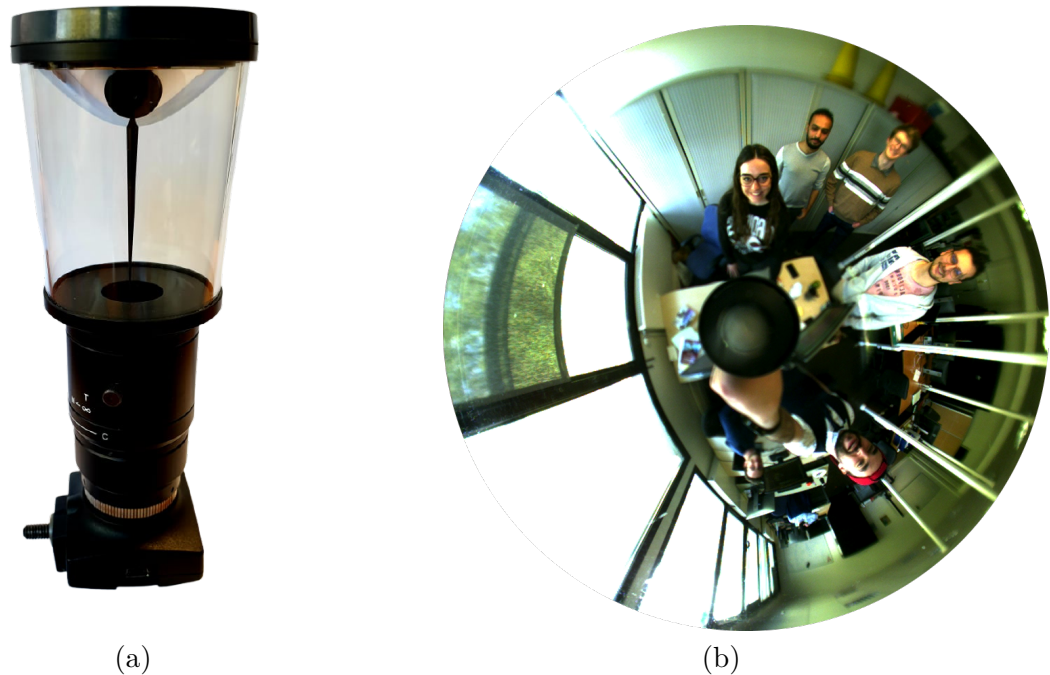


FIGURE 1.7 – (a) Système catadioptrique, (b) Image panoramique

sphérique [HONG et al., 1992], parabolique [STURM, 2002] ou à double courbure [SOUTHWELL et al., 1996], [FIALA et al., 2002]. Une telle combinaison, qualifiée de système catadioptrique, se retrouve dans le domaine de la robotique, en localisation [MARHIC et al., 1998] et en asservissement visuel [MEZOUAR et al., 2004]. C'est pourquoi nous proposons maintenant de nous intéresser aux systèmes catadioptriques.

1.2.2.1 Systèmes catadioptriques

Les systèmes catadioptriques combinent la réflexion et la réfraction des rayons lumineux.

Projeter une scène sur une surface réfléchissante est loin d'être une idée récente. M. C. Escher (1898-1972), dessinateur et graveur néerlandais, employait dans les années 1930 des miroirs sphériques réfléchissants pour observer une scène. À travers la gravure intitulée "Illustration 7" de 1934 (Fig. 1.8), l'artiste observe la scène réfléchi sur le miroir sphérique (Fig. 1.8 - b). Nos yeux, qui observent ce dernier, peuvent être assimilés à la caméra.

Le premier système catadioptrique employé en robotique a été développé par Yagi et al. [YAGI et al., 1990][YAGI et al., 1994], en combinant une caméra perspective avec un miroir conique. En 1995, un système avec le même type de miroir a été employé par Pegard et al. [PEGARD et MOUADDIB, 1996] pour la navigation d'un robot mobile.

Pour formaliser un modèle de projection simple, il est préférable que les systèmes catadioptriques aient un point de vue unique (*Single View Point*) : les droites liant les points de

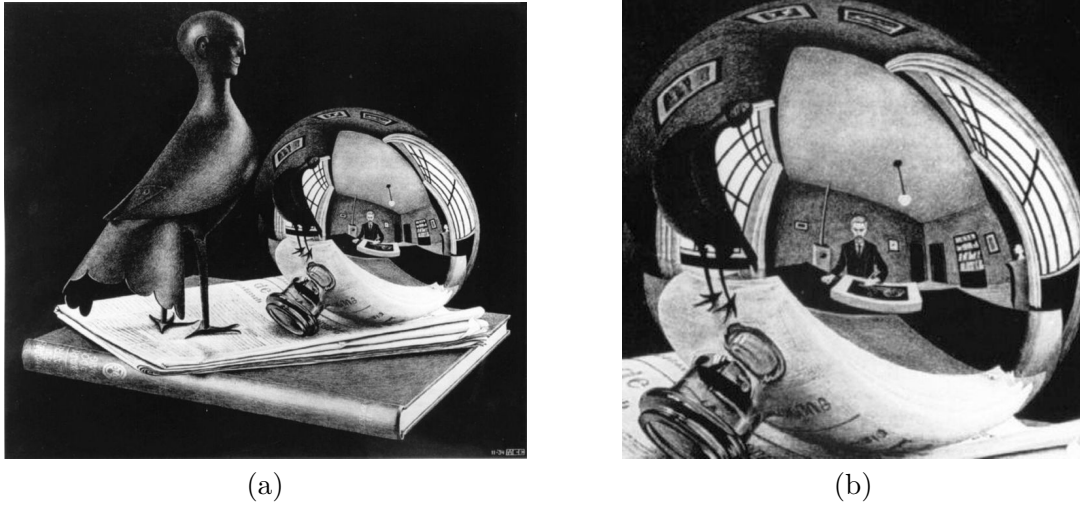


FIGURE 1.8 – (a) M. C. Escher, Nature morte à la sphère réfléchissante, lithographie, 1934, (b) zoom sur la surface réfléchissante.

l'espace et leurs projections sur le miroir passent toutes par un seul et unique point : le "point de vue unique". En pratique, il faut combiner un miroir parabolöide avec une caméra orthographique ou d'autres formes de miroirs (hyperbolique, etc...) avec une caméra perspective [BAKER et al., 1999] pour respecter cette contrainte. Geyer [GEYER et al., 2000] puis Barreto [BARRETO et al., 2001] ont proposé un modèle de projection stéréographique et ont classifié l'ensemble des systèmes catadioptriques centraux.

Enfin, la résolution de l'image panoramique acquise avec un système catadioptrique est variable spatialement et elle est moins bonne lorsque l'on se rapproche du centre de l'image. Cet effet est inversé pour les caméras *fish-eye* : la résolution est plus fine au voisinage du centre.

1.2.2.2 Projection stéréographique

Le modèle de projection stéréographique [GEYER et al., 2000, BARRETO et al., 2001] est basé sur une projection sphérique suivie par une projection sur un plan, à partir d'un second centre de projection appartenant à l'axe Z (Fig. 1.9). Dans le cas d'un miroir parabolöide, ce point se trouve sur le pôle Nord de la sphère. Dans le cas d'un miroir hyperbolöide, ce point se trouve entre le pôle Nord et le centre de la sphère.

La droite qui passe à la fois par un point 3D $\mathbf{M} = ({}^cX, {}^cY, {}^cZ)$ de la scène et le centre de la sphère (virtuelle) intercepte celle-ci au point ${}^s\mathbf{M} = ({}^sX, {}^sY, {}^sZ)$: ${}^sX = \lambda {}^cX$, ${}^sY = \lambda {}^cY$, ${}^sZ = \lambda {}^cZ$.

L'équation de la sphère est donnée par :

$${}^s X^2 + {}^s Y^2 + {}^s Z^2 = R^2 \quad (1.5)$$

On pose $R = 1$ sans perte de généralité et on remplace ${}^c X$, ${}^c Y$ et ${}^c Z$ dans 1.5 afin d'obtenir :

$$\lambda = \pm \frac{1}{\sqrt{{}^c X^2 + {}^c Y^2 + {}^c Z^2}} \quad (1.6)$$

Nous retenons la solution de λ qui permet d'obtenir le point sphérique du côté du point 3D. Ainsi, le point 3D $\mathbf{M} = ({}^c X, {}^c Y, {}^c Z, 1)^T$ exprimé dans le repère de la caméra ${}^c \mathcal{F}$ subit en premier lieu une projection sphérique Pr_s sur une sphère unitaire virtuelle :

$${}^s \mathbf{X} = Pr_s({}^c \mathbf{X}) \text{ avec, } \begin{cases} {}^s X = \frac{{}^c X}{\rho} \\ {}^s Y = \frac{{}^c Y}{\rho} \\ {}^s Z = \frac{{}^c Z}{\rho} \end{cases} \quad (1.7)$$

avec $\rho = \sqrt{{}^c X^2 + {}^c Y^2 + {}^c Z^2}$. Une projection perspective transforme ensuite le point de la sphère ${}^s \mathbf{X}$ en point \mathbf{x} , dans le plan image, avec $\mathbf{x} = Pr({}^s \mathbf{X})$. Le second centre de projection est donné par le centre du repère caméra : $(0, 0, \xi)^T$. ξ est la distance entre les deux centres de projection. L'association de ces deux projections peut être modélisée par $Pr_\xi(\cdot)$, une transformation directe de ${}^c \mathbf{X}$ vers \mathbf{x} , qui dépend de ξ , parfois appelé paramètre d'excentricité du miroir. Elle est donnée par :

$$\mathbf{x} = Pr_\xi({}^c \mathbf{X}) \text{ avec, } \begin{cases} x = \frac{{}^c X}{{}^c Z + \xi \rho} \\ y = \frac{{}^c Y}{{}^c Z + \xi \rho} \end{cases} \quad (1.8)$$

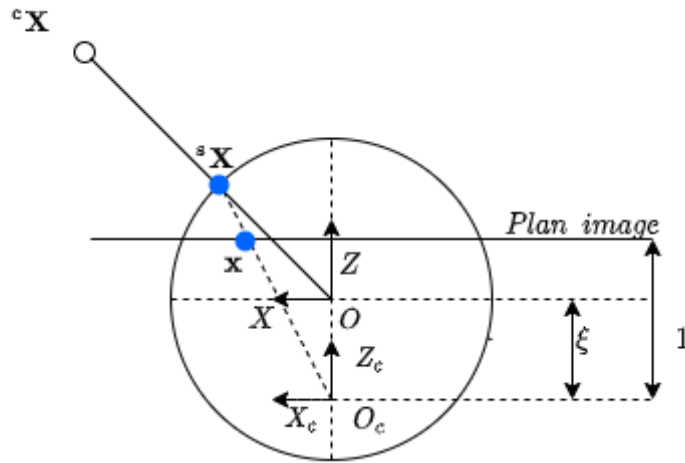


FIGURE 1.9 – Modèle de projection unifié avec une sphère virtuelle [CARON, 2010].

En agissant sur le paramètre ξ , il est possible de modéliser tous les systèmes catadioptriques centraux, chaque type de miroir étant caractérisé par son paramètre d'excentricité. En annulant ξ dans l'équation 1.8, on obtient le modèle d'une caméra perspective (Eq. 1.1).

Une transformation est enfin nécessaire pour représenter le point \mathbf{x} en coordonnées pixeliques \mathbf{u} . Ces coordonnées peuvent être calculées à l'aide de la matrice intrinsèque \mathbf{K} de l'équation 1.2. La projection omnidirectionnelle complète d'un point ${}^c\mathbf{X}$ vers un point donné en pixels ${}^p\mathcal{F}$ par le modèle de projection stéréographique s'écrit :

$$\mathbf{u} = \mathbf{K}Pr_{\xi}({}^c\mathbf{X}) \quad (1.9)$$

Comme la projection de la sphère vers le plan est inversible, il est possible de trouver les coordonnées sphériques d'un point à partir de ses coordonnées du plan image normalisé :

$${}^s\mathbf{X} = \mathbf{Pr}_{\xi}^{-1}(\mathbf{x}, \xi) \text{ avec, } \begin{cases} {}^sX = \frac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1}x \\ {}^sY = \frac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1}y \\ {}^sZ = -\xi + \frac{\xi + \sqrt{1 + (1 - \xi^2)(x^2 + y^2)}}{x^2 + y^2 + 1} \end{cases} \quad (1.10)$$

Bien que n'étant pas des caméras à point de vue unique, certains systèmes *fish-eye* peuvent être modélisés de la même manière, avec des résultats relativement proches de la projection réelle [HUGHES et al., 2010].

1.2.3 De l'objet à la caméra

Lorsque l'on modélise les caméras, on représente les points 3D dans le repère caméra ${}^c\mathcal{F}$. Cependant, l'objet est en général exprimé dans son propre repère orthonormé ${}^o\mathcal{F}$. La transformation rigide, aussi appelée "changement de repère", notée ${}^c\mathbf{M}_o$ permet de passer de l'un à l'autre. Elle est illustrée par la Figure 1.10.

L'orientation relative entre les deux repères est notée ${}^c\mathbf{R}_o$, avec ${}^c\mathbf{R}_o \in \mathbb{SO}(3)$. ${}^c\mathbf{t}_o$ est le vecteur de translation entre les deux repères, avec ${}^c\mathbf{t}_o \in \mathbb{R}^3$. La combinaison des transformations en position et en orientation forme une matrice homogène ${}^c\mathbf{M}_o$ associée à une transformation dite rigide, avec ${}^c\mathbf{M}_o \in \mathbb{SE}(3)$, donnée par l'équation suivante :

$${}^c\mathbf{M}_o = \begin{pmatrix} {}^c\mathbf{R}_o & {}^c\mathbf{t}_o \\ 0 & 1 \end{pmatrix}_{(4 \times 4)} \quad (1.11)$$

Grâce aux coordonnées homogènes, un point 3D ${}^o\mathbf{X} = ({}^oX, {}^oY, {}^oZ, 1)^T$ appartenant au

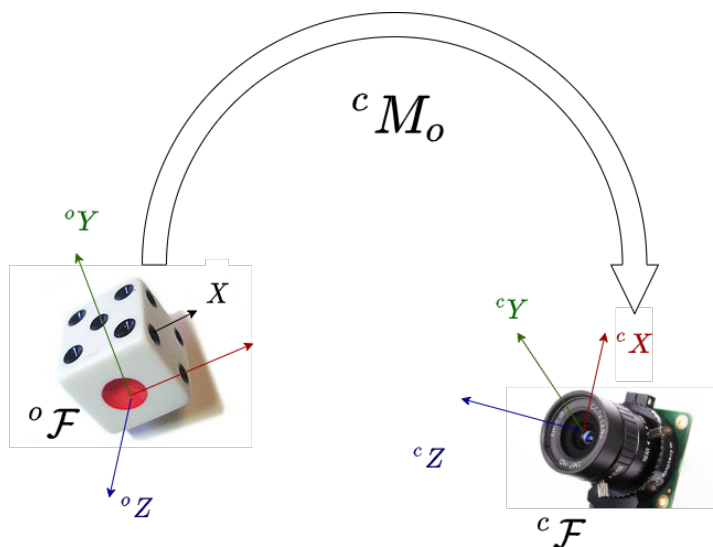


FIGURE 1.10 – Transformation rigide d'un point d'objet dans le repère caméra.

repère objet peut être exprimé dans le repère caméra comme suit :

$${}^c\mathbf{X} = {}^c\mathbf{M}_o {}^o\mathbf{X}. \quad (1.12)$$

La matrice de rotation ${}^c\mathbf{R}_o$ s'exprime selon différents formalismes : par exemple celui d'Euler, le formalisme axe/angle, ou encore la représentation par quaternions. La combinaison de l'équation 1.12 avec l'équation 1.4 permet de construire le modèle de la projection perspective d'un point de l'objet :

$$\mathbf{u} = \mathbf{KPr}({}^c\mathbf{M}_o {}^o\mathbf{X}). \quad (1.13)$$

La combinaison de l'équation 1.12 avec l'équation 1.9, permet quant à elle de construire le modèle de la projection catadioptrique d'un point de l'objet :

$$\mathbf{u} = \mathbf{KPr}_\xi({}^c\mathbf{M}_o {}^o\mathbf{X}). \quad (1.14)$$

1.3 Différentes approches de l'asservissement visuel

L'un des problèmes clé en asservissement visuel est la façon de comparer les images courante et désirée. Deux classes d'approches permettent de le faire : les approches basées sur les primitives géométriques extraites des images et les approches photométriques qui utilisent

l'ensemble des pixels.

1.3.1 Asservissement visuel basé primitives géométriques

Ce type d'approches utilise des mesures géométriques extraites de l'image désirée et de celles acquises tout au long de l'asservissement visuel. Il s'agit, par exemple, des coordonnées de points [CHAUMETTE et al., 2006], d'équation de droite [ANDREFF et al., 2002] ou de cercle [ESPIAU et al., 1992] [CHAUMETTE, 1990]. Les primitives géométriques peuvent être utilisées de différentes manières pour calculer la loi de commande.

Dans le cadre de l'asservissement visuel basé image, les mesures géométriques calculées avec des éléments du plan image sont injectées dans la loi de commande qui a pour tâche de réguler l'erreur. Dans le cadre de l'asservissement visuel basé pose des primitives 2D sont utilisées afin d'estimer la pose de la caméra par rapport à la scène observée. La régulation de l'erreur nécessite la connaissance du modèle 3D de la scène.

Concentrons nous maintenant sur l'asservissement visuel basé image. L'écart entre les primitives courantes et les primitives désirées est alors modélisé dans l'image par l'erreur suivante :

$$\mathbf{e}(\mathbf{r}) = \mathbf{s}(\mathbf{r}) - \mathbf{s}^* \quad (1.15)$$

avec :

- \mathbf{r} est un vecteur qui contient les positions et les orientations de la caméra :
 $\mathbf{r} = (t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$,
- \mathbf{s} est un vecteur qui contient les primitives géométriques extraites de l'image courante prise par la caméra à la pose courante \mathbf{r} ,
- \mathbf{s}^* contient les primitives géométriques fixes extraites de l'image désirée. Autrement dit \mathbf{s}^* est considérée comme la consigne de la tâche.

Le torseur cinématique de la caméra $\mathbf{v} \in \mathfrak{se}(3)$ peut être obtenu en dérivant le vecteur pose $\mathbf{r} : \dot{\mathbf{r}} = (v, w) = \dot{\mathbf{r}}$, où $v = (v_x, v_y, v_z)$, $w = (w_x, w_y, w_z)$ représentent respectivement les vitesses linéaires et angulaires du torseur cinématique de la caméra avec $\mathfrak{se}(3) \simeq \mathbb{R}^3 \times \mathbb{R}^3$ l'espace tangent à $\mathbb{SE}(3)$. \mathbf{s} est différentiable et le déplacement des primitives géométriques $\dot{\mathbf{s}}$ est lié aux vitesses \mathbf{v} du déplacement de la caméra [ESPIAU et al., 1992] par la relation suivante :

$$\dot{\mathbf{s}} = \frac{d\mathbf{s}}{dt} = \frac{\partial \mathbf{s}}{\partial \mathbf{r}} \frac{d\mathbf{r}}{dt} = \frac{\partial \mathbf{s}}{\partial \mathbf{S}} \frac{\partial \mathbf{S}}{\partial \mathbf{r}} \frac{d\mathbf{r}}{dt} = \mathbf{L}_s \mathbf{v}, \quad (1.16)$$

où $\mathbf{L}_s \in \mathbb{R}^{k \times 6}$ représente la matrice qui relie le déplacement des k primitives aux mouvements de la caméra. Cette matrice est connue dans la littérature sous le nom de matrice d'interaction [CHAUMETTE, 1990]. Elle est la composition de deux jacobiniennes. La première lie les primitives images aux primitives 3D exprimées dans le repère caméra et dépend directement du modèle

de projection. La seconde est indépendante du modèle de projection. La matrice d'interaction en vision perspective a été initialement exprimée pour des primitives de type points [FEDDEMA et al., 1989], et ensuite pour des primitives plus complexes comme des droites, des segments, des cercles, des cylindres et des sphères [CHAUMETTE, 1990].

L'apparition du modèle de projection unifié [BARRETO et al., 2001] a permis le développement de l'asservissement visuel pour la vision omnidirectionnelle et les matrices d'interaction ont été exprimées pour les différents types de primitives : les droites [BARRETO et al., 2003] [HADJ-ABDELKADER, 2006] et les sphères [FOMENA, 2008].

Pour établir la loi de commande, on impose (par exemple) une décroissance exponentielle de l'erreur :

$$\dot{\mathbf{e}} = -\mu\mathbf{e}, \quad (1.17)$$

avec μ un gain scalaire pour régler la vitesse de convergence tout en conservant la stabilité du système.

A partir des équations 1.15 et 1.16, nous pouvons relier la variation temporelle de l'erreur au déplacement de la caméra :

$$\dot{\mathbf{e}} = \mathbf{L}_s \mathbf{v} \quad (1.18)$$

Nous injectons 1.17 dans 1.18 afin de définir la loi de commande de l'asservissement visuel :

$$\mathbf{v} = -\mu\mathbf{L}_s^+ \mathbf{e}, \quad (1.19)$$

où \mathbf{L}_s^+ est la pseudo-inverse de la matrice d'interaction. Même si la matrice d'interaction est analytiquement connue, elle dépend aussi d'autres paramètres, tels que les paramètres intrinsèques de la caméra. Ces paramètres sont estimés au moment de l'étalonnage de la caméra. Pour cette raison, nous travaillons avec une matrice d'interaction estimée $\hat{\mathbf{L}}_s$. La loi de commande de l'asservissement visuel s'écrit alors :

$$\mathbf{v} = -\mu\hat{\mathbf{L}}_s^+ \mathbf{e}. \quad (1.20)$$

La matrice d'interaction des capteurs à point de vue unique qui lie les variations des vitesses de ces capteurs aux variations du vecteur de coordonnées normalisées \mathbf{x} associées au plan image omnidirectionnelle a été établie par [HADJ-ABDELKADER, 2006] :

$$\mathbf{L}_x = \begin{bmatrix} -\frac{1+x^2(1-\xi(\alpha+\xi))+y^2}{\rho(\alpha+\xi)} & \frac{\xi xy}{\rho} & \frac{\alpha x}{\rho} & xy & -\frac{(1+x^2)\alpha-\xi y^2}{\alpha+\xi} & y \\ \frac{\xi xy}{\rho} & -\frac{1+y^2(1-\xi(\alpha+\xi))+x^2}{\rho(\alpha+\xi)} & \frac{\alpha y}{\rho} & \frac{(1+y^2)\alpha-\xi x^2}{\alpha+\xi} & -xy & -x \end{bmatrix} \quad (1.21)$$

Avec :

$$\begin{aligned} - \alpha &= \sqrt{1 + (1 - \xi^2)(x^2 + y^2)} \\ - \rho &= \sqrt{cX^2 + cY^2 + cZ^2} \end{aligned}$$

Nous pouvons obtenir la matrice d'interaction des caméras perspectives seulement en mettant la valeur de l'excentricité du miroir $\xi = 0$.

1.3.2 Asservissement visuel photométrique

La détection et le suivi de primitives géométriques (points, lignes droites, pose, homographie, etc.) est parfois difficile [E. MARCHAND et al., 2005], encore plus dans le cas d'images issues de système catadioptrique à cause des distorsions, notamment. L'asservissement visuel photométrique consiste à utiliser l'image dans sa globalité, ce qui permet d'éliminer le problème de la détection et du suivi.

Dans les travaux de Collewet et Marchand [COLLEWET et al., 2008a], les auteurs proposent d'introduire l'intensité de tous les pixels de l'image désirée et l'image courante dans la loi de commande.

Caron *et al.* [CARON et al., 2010] ont étendu cette approche à la vision omnidirectionnelle en profitant du champ visuel important du système catadioptrique ou *fish-eye*.

1.3.2.1 Formulation du problème

L'asservissement visuel est dit purement photométrique quand l'erreur s'exprime directement avec les intensités de l'image prise par la caméra à la pose désirée \mathbf{r}^* et les intensités de l'image courante prises à la pose \mathbf{r} . La fonction d'erreur devient alors :

$$\mathbf{e}(\mathbf{r}) = \mathbf{I}(\mathbf{r}) - \mathbf{I}^*(\mathbf{r}^*) \quad (1.22)$$

Les vecteurs \mathbf{I}^* et \mathbf{I} contiennent respectivement les $(P \times M)$ intensités des pixels de l'image désirée et les $(P \times M)$ intensités des pixels de l'image courante à la pose \mathbf{r} .

Nous pouvons définir le problème de l'asservissement visuel purement photométrique comme étant un problème d'optimisation où l'objectif est de minimiser la fonction de coût SSD (*Sum of Squared Differences*) :

$$\mathcal{C}(\mathbf{r}) = \frac{1}{2} \|\mathbf{e}(\mathbf{r})\|^2 \quad (1.23)$$

Pour minimiser ce coût il faut lier la variation de l'intensité au mouvement dans l'image grâce au flot optique.

1.3.2.2 Contrainte du flot optique et gradient image

Le flot optique peut être décrit comme étant un mouvement de l'objet dans l'image entre deux acquisitions consécutives (Fig 1.11). Nous faisons l'hypothèse que la lumière réfléchi par un point de la scène reste constante indépendamment du mouvement relatif caméra/scène.

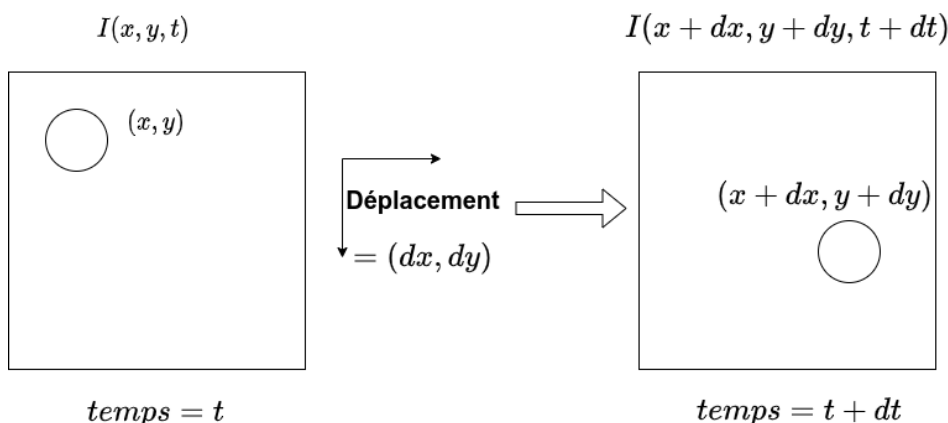


FIGURE 1.11 – Modélisation du flot optique.

Afin d'estimer le flot optique entre deux images consécutives, nous représentons l'intensité de l'image I en fonction de $\mathbf{x} = (x, y)$ et du temps (t). Nous supposons que si on déplace chaque pixel d'une image de $d\mathbf{x} = (dx, dy)$ durant un temps (dt), nous obtenons la nouvelle image $I(\mathbf{x} + d\mathbf{x}, t + dt)$:

$$I(\mathbf{x} + d\mathbf{x}, t + dt) = I(\mathbf{x}, t). \quad (1.24)$$

Nous appliquons un développement de Taylor d'ordre 1 de $I(\mathbf{x} + d\mathbf{x}, t + dt)$:

$$I(x + dx, y + dy, t + dt) \simeq I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt \quad (1.25)$$

et comme $I(x + dx, y + dy, t + dt) = I(x, y, t)$, on obtient la contrainte du flot optique [HORN et al., 1981] :

$$\frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt = 0, \quad (1.26)$$

qui donne, sous forme matricielle :

$$\nabla \mathbf{I}^T \dot{\mathbf{x}} + \mathbf{I}_t = 0, \quad (1.27)$$

avec $\nabla \mathbf{I}$ le gradient spatial de l'image \mathbf{I} au point (x, y) ; et $\mathbf{I}_t = \frac{\partial \mathbf{I}}{\partial t}$ le gradient temporel.

Par ailleurs, le déplacement d'un point \mathbf{x} de l'image est lié au mouvement de la caméra par $\dot{\mathbf{x}} = \mathbf{L}_x \mathbf{v}$ [COLLEWET et al., 2008b], (\mathbf{L}_x est la matrice d'interaction d'un point cf. Eq. 1.21). L'équation 1.27 s'exprime alors après remplacement de $\dot{\mathbf{x}}$ par :

$$\mathbf{I}_t = -\nabla \mathbf{I}^T \mathbf{L}_x \mathbf{v} \quad (1.28)$$

Pour la suite, on pose \mathbf{L}_I la matrice d'interaction qui relie l'intensité du point \mathbf{x} aux mouvements de la caméra, comme étant :

$$\mathbf{L}_I = -\nabla \mathbf{I}^T \mathbf{L}_x \quad (1.29)$$

Afin d'établir \mathbf{L}_I , il faut calculer $\nabla \mathbf{I}$. Rappelons que :

$$\nabla \mathbf{I} = [\nabla I_x \quad \nabla I_y]^T = \left[\frac{\partial I}{\partial x} \quad \frac{\partial I}{\partial y} \right]^T \quad (1.30)$$

Les gradients images peuvent être approchés à partir de l'image originale en utilisant la convolution (*) entre l'image \mathbf{I} et deux filtres dérivatifs dont les coefficients correspondent aux dérivées d'un filtre gaussien [É. MARCHAND et al., 2005] :

$$\frac{\partial I}{\partial x} = \frac{1}{8418} \begin{bmatrix} -112 & -913 & -2047 & 0 & 2047 & 913 & 112 \end{bmatrix} * \mathbf{I} \quad (1.31)$$

$$\frac{\partial I}{\partial y} = \frac{1}{8418} \begin{bmatrix} -112 & -913 & -2047 & 0 & 2047 & 913 & 112 \end{bmatrix}^T * \mathbf{I} \quad (1.32)$$

La figure 1.12 montre un résultat de gradient dans la direction x (Fig. 1.12 - b) et y (Fig. 1.12 - c).

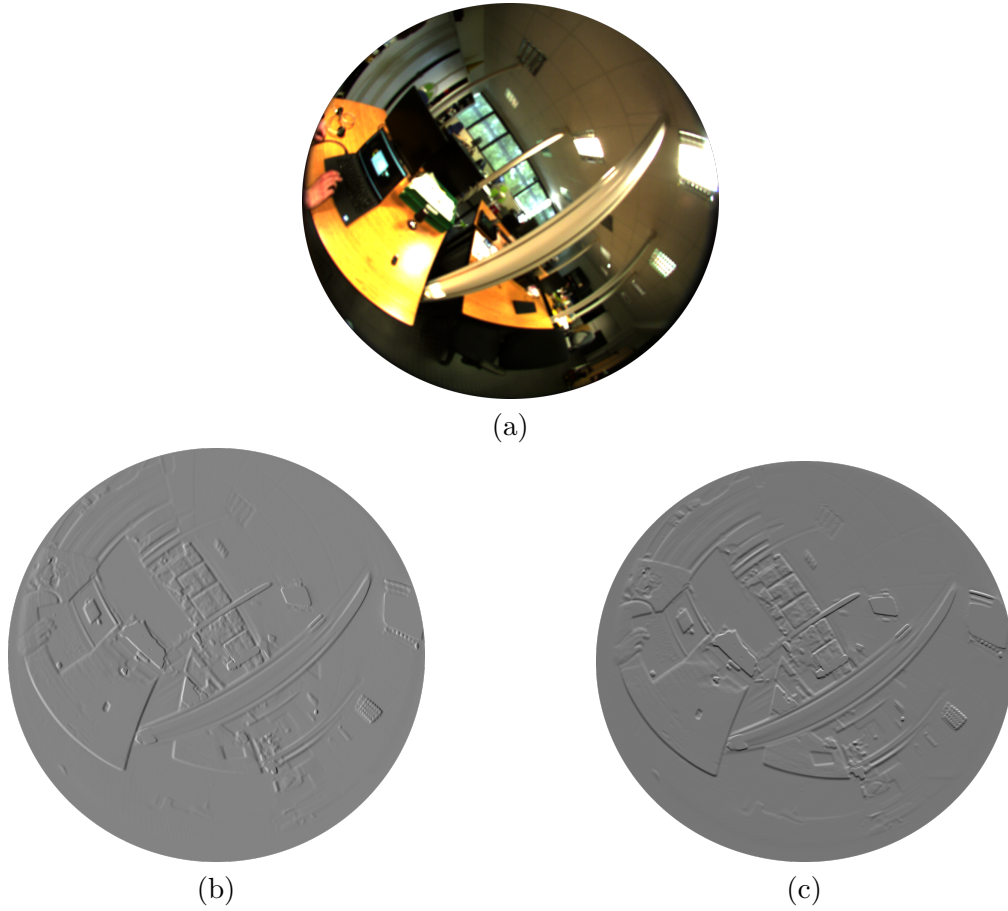


FIGURE 1.12 – Gradient : (a) image d'intensité, (b) gradient selon la direction de x, (c) gradient selon la direction de y.

1.3.2.3 Loi de commande de l'asservissement visuel photométrique

La convergence de la caméra vers la pose désirée peut être vue comme étant une solution d'un problème d'optimisation non-linéaire. Afin de déterminer la loi de commande, plusieurs méthodes d'optimisation non-linéaire peuvent être utilisées. Par exemple, avec la méthode Gauss-Newton, on obtient :

$$\mathbf{v} = -\mu \mathbf{L}_I^+ (\mathbf{I}(\mathbf{r}) - \mathbf{I}(\mathbf{r}^*)) \quad (1.33)$$

où \mathbf{L}_I est la matrice d'interaction construite avec toutes les matrices d'interactions (Eq. 1.29) de chaque pixel de l'image en les empilant :

$$\mathbf{L}_I = \begin{pmatrix} \mathbf{L}_I(1) \\ \mathbf{L}_I(2) \\ \vdots \\ \mathbf{L}_I(P \times M) \end{pmatrix} \quad (1.34)$$

Dans [COLLEWET et al., 2008a], les auteurs ont utilisé la loi de commande de type Levenberg-Marquardt (LM). Cette dernière est utilisée lorsque la jacobienne n'est pas de rang plein (ou proche de l'être) [BERGOU et al., 2018] :

$$\mathbf{v} = -\mu(\mathbf{H} + \beta \text{diag}(\mathbf{H}))^{-1} \mathbf{L}_I^T (\mathbf{I}(\mathbf{r}) - \mathbf{I}(\mathbf{r}^*)) \quad (1.35)$$

avec $\mathbf{H} = \mathbf{L}_I^T \mathbf{L}_I$. Le paramètre β est à régler. Si ce dernier est élevé, la loi de commande 1.35 se comporte comme une descente de gradient alors qu'une faible valeur de β conduit l'équation 1.35 à un comportement type Gauss-Newton (Eq. 1.33).

Dans ce chapitre et le suivant, nous utiliserons la loi de commande basée Gauss-Newton (Eq. 1.33) car elle évite le réglage du β de la loi de commande Levenberg-Marquardt.

1.4 Application de l'asservissement visuel photométrique omnidirectionnel dans un environnement virtuel

Dans le reste de ce chapitre, nous allons appliquer l'asservissement visuel photométrique omnidirectionnel dans un environnement virtuel afin de faciliter l'évaluation de la précision et des taux de convergence (nombre de succès sur nombre total de tentatives).

Nous utilisons la même matrice d'interaction et le même calcul de la vitesse de la caméra mais pour déplacer la caméra virtuelle. C'est sa pose, définie sur le groupe $\text{SE}(3)$, qui est mise à jour lors de chaque itération *it* grâce à l'application exponentielle de l'algèbre de Lie de $\text{SE}(3)$ [MA et al., 2004] :

$${}^c \mathbf{M}_o^{it} = {}^c \mathbf{M}_o^{it-1} e^{[\mathbf{v}]} \quad (1.36)$$

L'environnement visuel est constitué dans notre étude par un nuage de points de quatre rues de la ville d'Amiens en France². Le nuage de points réunit treize scans 3D avec des couleurs photographiques Rouge-Vert-Bleu, tous obtenus avec un scanner Lidar Faro Focus 3D. La caméra virtuelle simule le modèle unifié des caméras (voir Section 1.2.2.2) implémentée avec un vertex shader dans le logiciel Unity 3D (<https://unity.com/fr>). Il est relié à notre programme C++ d'asservissement visuel photométrique omnidirectionnel qui utilise la

2. <https://www.google.com/maps/place/Cathédrale+Notre-Dame+d'Amiens/@49.8958793,2.2994662,18.25z>



(a)

(b)

FIGURE 1.13 – (a) : Environnement réel (vue perspective), (b) : Rendu omnidirectionnel de l'environnement utilisé sur Unity pour l'asservissement visuel

bibliothèque ViSP (*Visual Servoing Platform* <https://visp.inria.fr/>) [É. MARCHAND et al., 2005]. Les paramètres intrinsèques de la caméra correspondent à ceux d'une vraie caméra catadioptrique calibrée préalablement par le logiciel Hyscas [CARON et al., 2011].

La figure 1.13 - b montre le rendu catadioptrique au sein de cet environnement virtuel et la figure 1.13 - a montre la même rue (*Rue Robert de Luzarches, Amiens*) réelle.

1.4.1 Protocole

Dans un premier temps, nous allons asservir la vitesse en simulation sur un seul Degré De Liberté (DDL). Ensuite, des DDL supplémentaires seront ajoutés au fur et à mesure pour rendre la simulation plus réaliste. Ce protocole progressif permet entre autres de :

- tracer facilement une évaluation de la fonction de coût afin de déduire le domaine de convergence,
- étudier l'asservissement visuel en dehors du couplage qui existe entre les différents DDL : par exemple entre les translations selon \vec{X} et \vec{Y} et les rotations selon les mêmes axes.

Grâce à la simulation, nous pouvons facilement obtenir la profondeur associée à chaque point 3D, ce qui est bien plus complexe dans la réalité. Nous précisons que le ciel est ignoré dans la loi de commande (absence de point 3D rendu).

Afin d'évaluer expérimentalement le domaine de convergence, la caméra virtuelle est placée en différentes poses désirées dans l'environnement virtuel. Les vitesses calculées lors de l'asservissement visuel sont directement appliquées pour la mise à jour de la pose de la ca-

méra virtuelle. La valeur du gain de convergence μ est réglée à 0.2 pour tous les résultats. Cette valeur est choisie faible afin d'éviter les oscillations à convergence et fixe pour pouvoir comparer les différents résultats.

1.4.2 Résultats

Cette section présente des résultats d'asservissements visuels photométriques afin d'en montrer le comportement et les limites.

1.4.2.1 Simulation en 1 DDL

Nous traçons d'abord la fonction de coût (Eq. 1.23) en faisant une translation pure de la caméra virtuelle selon l'axe \vec{X} de -30 m à $+30$ m autour d'une pose désirée. La fonction de coût est calculée par la SSD des images prises par la caméra à chaque pose avec l'image désirée \mathbf{I}^* . Nous pouvons remarquer l'existence de minima locaux qui limitent le domaine de convergence (Fig. 1.14) entre -2 m à $+2$ m.

Nous fixons ensuite la pose initiale de la caméra virtuelle \mathbf{r}_0 par une translation pure selon l'axe \vec{X} égale à $1,85$ m de la pose désirée \mathbf{r}^* . L'image désirée \mathbf{I}^* correspond à l'image omnidirectionnelle obtenue à la pose \mathbf{r}^* (Fig. 1.15 - a). L'image initiale \mathbf{I} obtenue à la pose \mathbf{r}_0 est montrée en figure 1.15 - b.

L'asservissement visuel purement photométrique a réussi à converger vers la pose désirée ((Fig. 1.15 - c), (Fig. 1.15 - e)) au bout de 206 itérations. L'erreur entre l'image désirée et courante, calculée à partir des différentes images obtenues par la caméra catadioptrique virtuelle à chaque pose, décroît au fil des itérations et tend vers 0^3 (Fig. 1.15 - d). Nous pouvons remarquer la précision submillimétrique après 1000 itérations, de l'ordre de $1,5e-8$ m, de l'asservissement visuel purement photométrique.

1.4.2.2 Simulation en 2 DDL (deux translations)

Afin d'élargir notre champ d'étude, nous proposons d'ajouter un DDL au processus d'optimisation qui s'effectue sur les deux axes \vec{X} et \vec{Y} .

Nous suivons ensuite les mêmes étapes que dans l'exemple précédent, cette fois-ci avec une translation de la caméra selon les axes \vec{X} et \vec{Y} de -4 m à $+4$ m autour de la pose désirée avec un pas régulier de 10 cm pour respecter les limites géométriques du modèle 3D.

La fonction de coût en 2D est montrée en figure 1.16. Elle contient quelques minima locaux.

3. Les intensités de pixels étant codées entre 0 et 255, même une erreur minimum de $0,1$ mm donne l'apparence d'une valeur importante d'erreur photométrique ($\mathcal{C}(\mathbf{r}) = 1000$ est en fait faible).

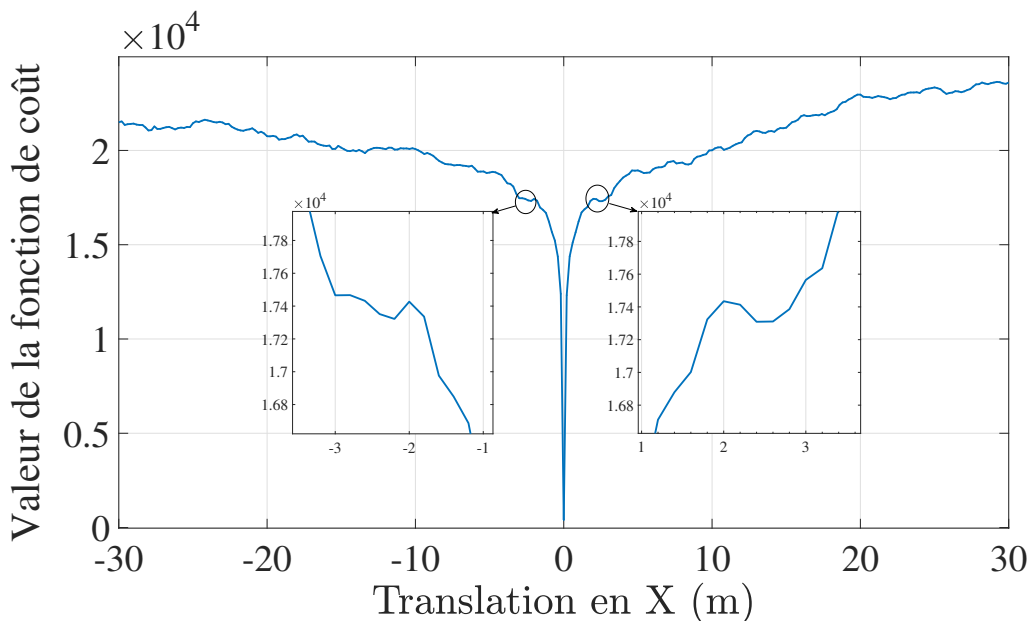


FIGURE 1.14 – Fonction de coût pour un DDL.

En respectant le domaine de convergence de la fonction de coût, la pose initiale de la caméra est fixée par une translation de -2 m selon l'axe \vec{X} et de -1 m selon l'axe \vec{Y} . La figure 1.17 - b représente l'image initiale $\mathbf{I}(\mathbf{r}_0)$, soit la vue de la caméra virtuelle à la pose initiale \mathbf{r}_0 .

Nous observons que la caméra converge de la pose initiale \mathbf{r}_0 vers la pose désirée \mathbf{r}^* . La figure 1.17 - c montre la différence entre l'image désirée \mathbf{I}^* (Fig. 1.17 - a) et l'image enregistrée à la pose finale $\mathbf{I}(\mathbf{r}_f)$ en 1000 itérations. L'erreur de positionnement à la fin de l'asservissement $[\Delta_X, \Delta_Y]$ est de $[-4,5e-5 \text{ m}, -6,8e-5 \text{ m}]$ avec une norme de $8,2e-5 \text{ m}$. Nous pouvons remarquer à nouveau la précision submillimétrique.

Pour la même pose désirée (Fig. 1.17 - a), nous effectuons une translation importante en dehors du domaine de convergence, -10 m selon l'axe \vec{X} et -4 m selon l'axe \vec{Y} . Le processus d'optimisation tombe dans un minimum local après seulement 6 itérations (Fig. 1.18) et l'erreur de positionnement est $[\Delta_X, \Delta_Y] = [-9.99 \text{ m}, -3.94 \text{ m}]$, en accord avec la surface de la fonction de coût. L'exemple que nous venons de proposer s'inscrit dans une démarche comparative pour le prochain chapitre de contribution.

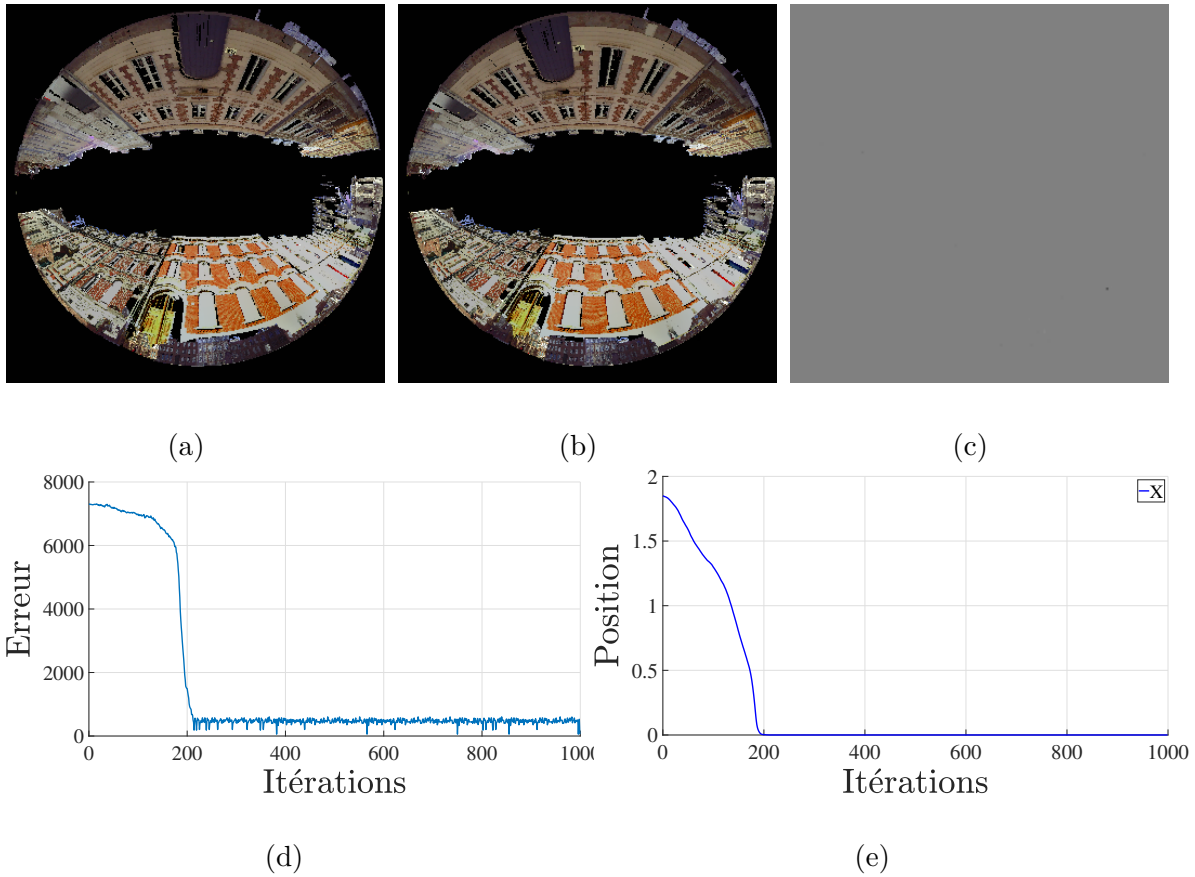


FIGURE 1.15 – Simulation en 1-DDL, erreur initiale : $\Delta_X = 1.8$ m est : (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l'évolution de l'erreur et (e) l'évolution de la position de la caméra.

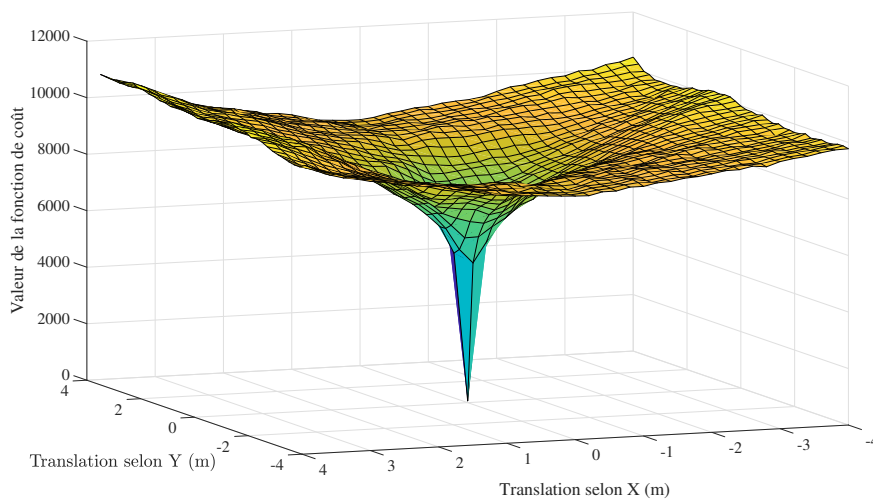


FIGURE 1.16 – Fonction de coût pour deux DDL.

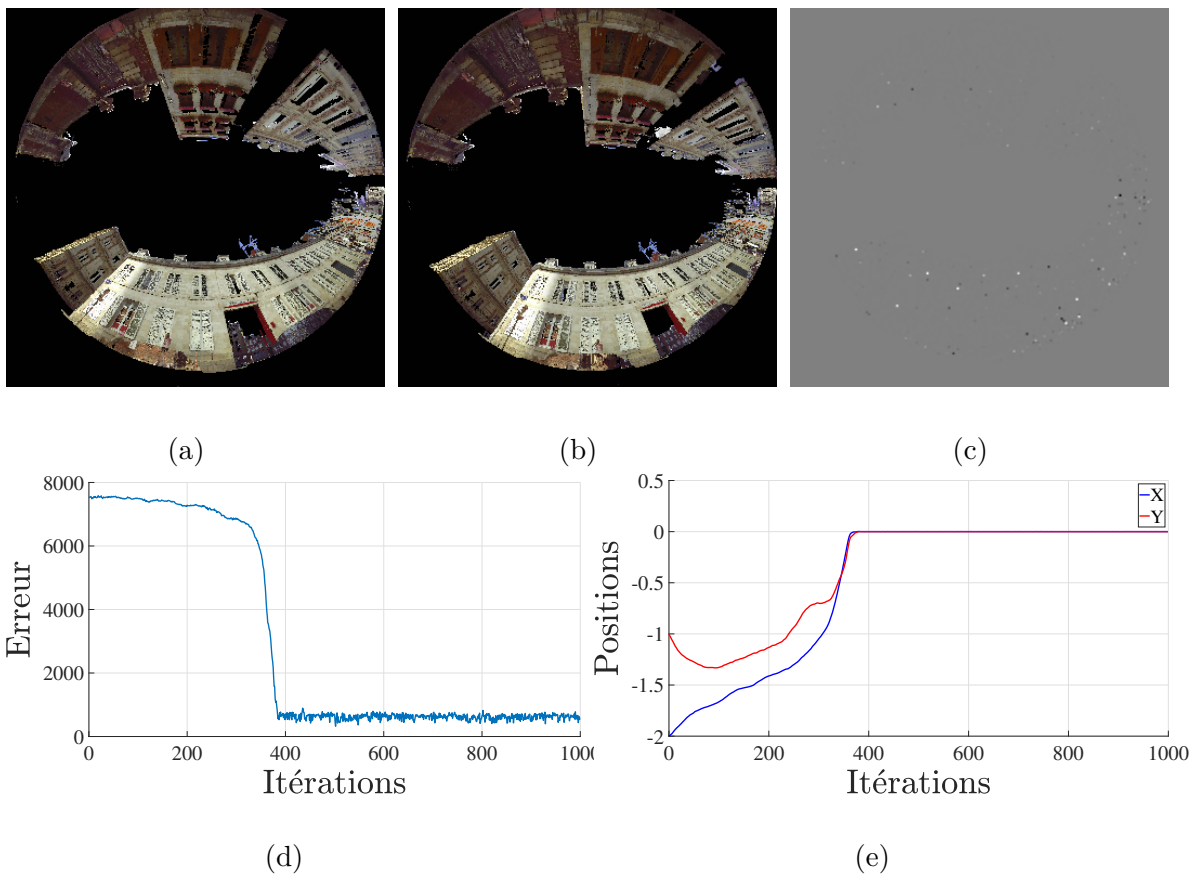


FIGURE 1.17 – Simulation en 2-DDL, erreur initiale : $[\Delta_X, \Delta_Y] = [-2 \text{ m}, -1 \text{ m}]$, (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l'évolution de l'erreur et (e) l'évolution de la position de la caméra.

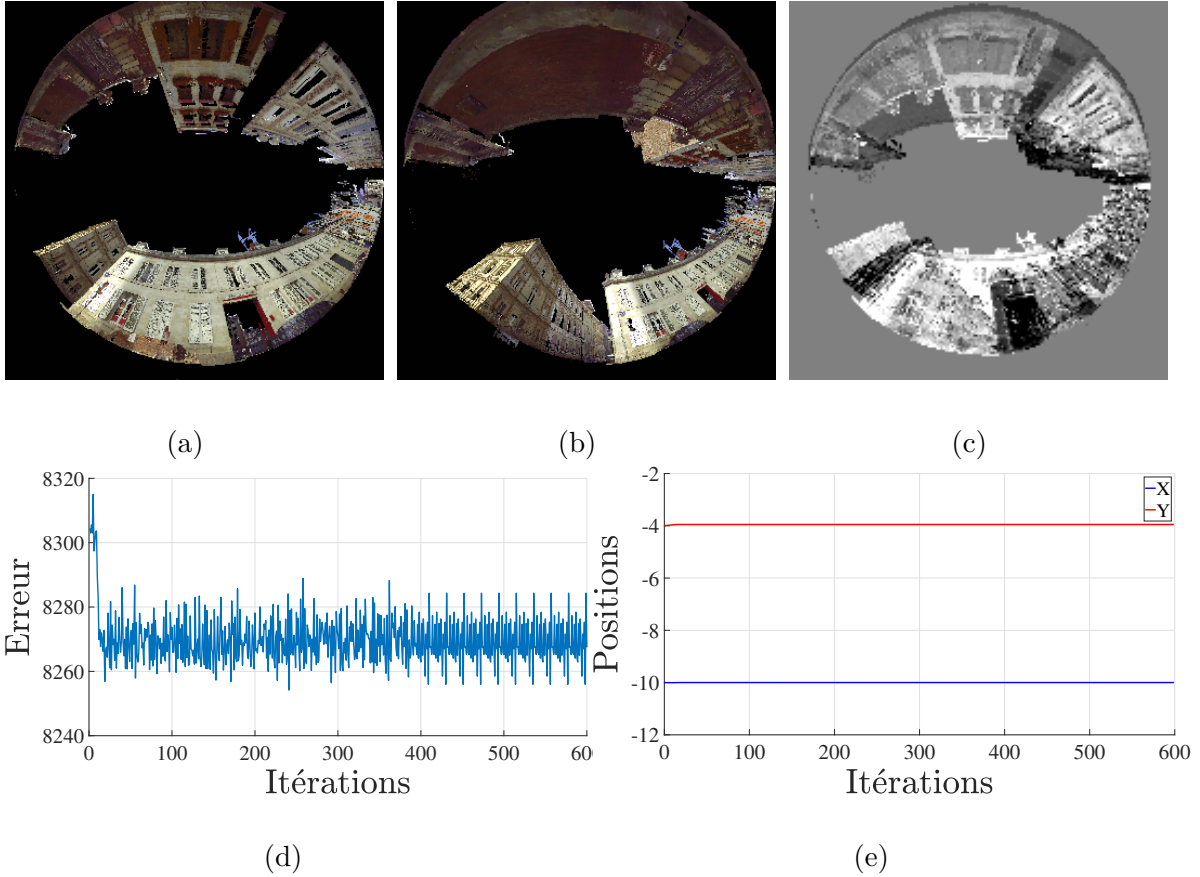


FIGURE 1.18 – Simulation en 2-DDL bis, erreur initiale : $[\Delta_X, \Delta_Y] = [-10 \text{ m}, -4 \text{ m}]$: (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l'évolution de l'erreur et (e) l'évolution de la position de la caméra.

1.4.2.3 Simulation en 3 DDL (3 translations)

— Simulation 1 :

Dans cette expérience, nous activons les 3 DDL de translation. Nous introduisons les erreurs initiales de position : $[\Delta_X, \Delta_Y, \Delta_Z] = [1 \text{ m}, -1 \text{ m}, -1 \text{ m}]$ afin d'obtenir la pose initiale \mathbf{r}_0 représentée par la figure 1.19 - a.

Nous enregistrons une convergence de la caméra virtuelle vers la pose désirée. La figure 1.19 - c montre la différence entre l'image finale $\mathbf{I}(\mathbf{r}_f)$ et l'image désirée \mathbf{I}^* . L'erreur de position décroît et tend vers 0. L'erreur finale de l'asservissement visuel est $[-2, 3e - 4 \text{ m}, 2, 1e - 4 \text{ m}, 2, 2e - 4 \text{ m}]$, ce qui donne une norme de position submillimétrique de $3, 8e - 4 \text{ m}$.

— Simulation 2 :

Pour la même pose désirée (Fig. 1.19 - a), nous effectuons une translation de la caméra virtuelle depuis la pose désirée, de -3 m selon l'axe \vec{Z} , de -2 m selon l'axe \vec{X} et de -1 m selon

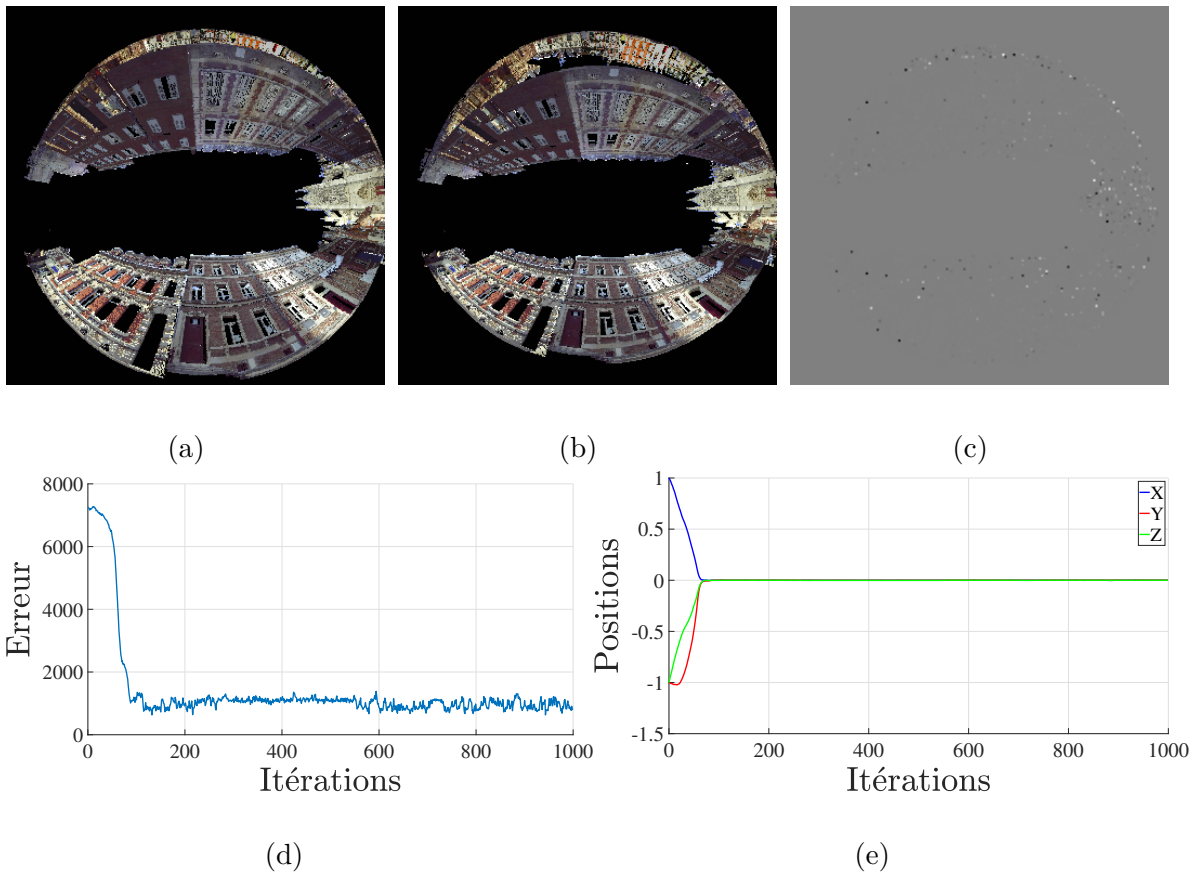


FIGURE 1.19 – Simulation en 3-DDL : erreur initiale : $[\Delta_X, \Delta_Y, \Delta_Z] = [1 \text{ m}, -1 \text{ m}, -1 \text{ m}]$, (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l'évolution de l'erreur et (e) l'évolution de la position de la caméra.

l'axe \vec{Y} .

Nous pouvons observer que le processus d'optimisation converge vers un minimum local, ce qui signifie que la pose initiale de la caméra se trouve en dehors du domaine de convergence. La figure 1.20 - c montre des différences entre l'image désirée et la pose "finale" (quand l'asservissement visuel est bloqué dans le minimum local). Nous enregistrons une erreur de positionnement finale $[\Delta_X, \Delta_Y, \Delta_Z]$ égale à $[-0,6 \text{ m}, -1,8 \text{ m}, 3,7 \text{ m}]$.

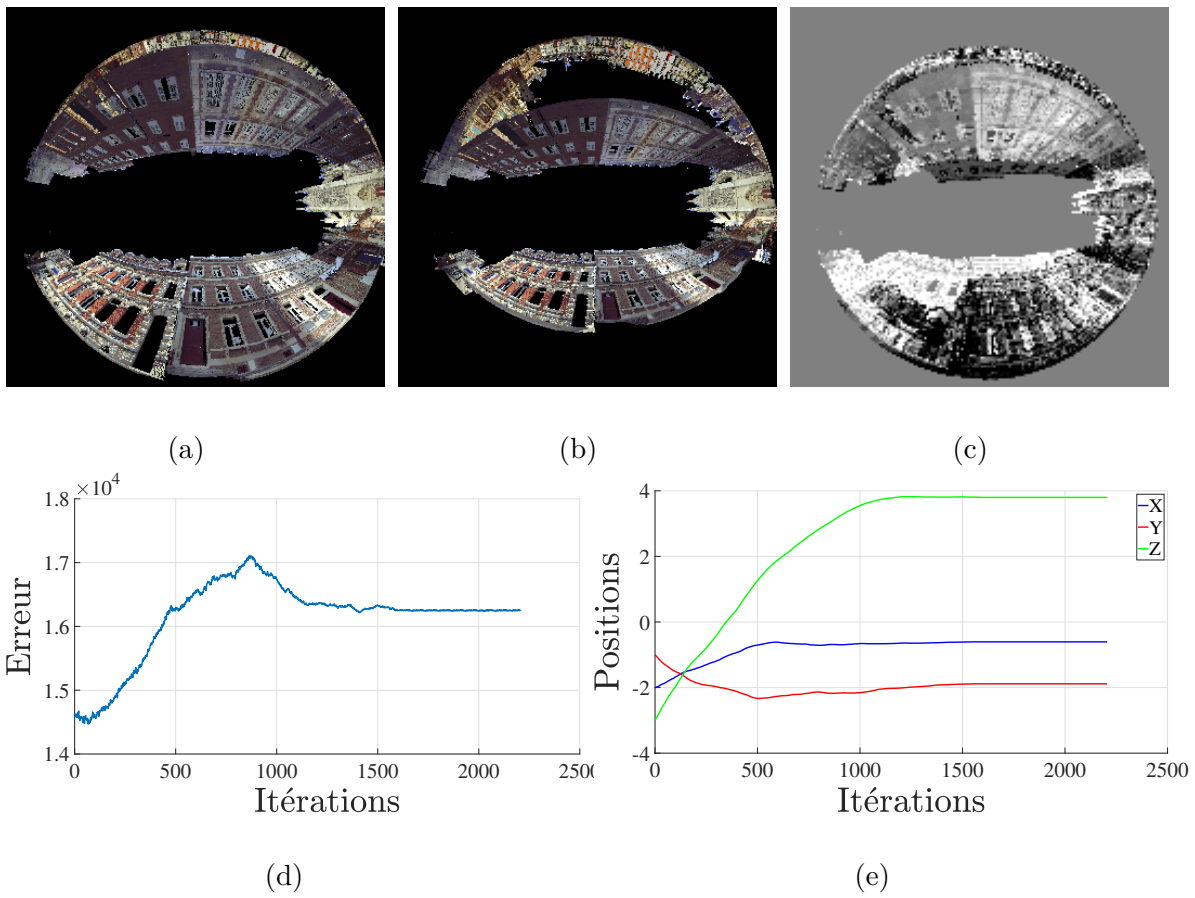


FIGURE 1.20 – Simulation en 3-DDL : erreur initiale : $[\Delta_X, \Delta_Y, \Delta_Z] = [-2 \text{ m}, -1 \text{ m}, -3 \text{ m}]$, (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l'évolution de l'erreur et (e) l'évolution de la position de la caméra.

1.4.2.4 Simulation en 6 DDL (3 translations et 3 rotations)

— Simulation 1 :

Dans cette expérience, nous activons les 6 DDL durant l'optimisation. Nous introduisons alors une erreur sur tous les DDL afin d'obtenir la pose initiale \mathbf{r}_0 représentée par la figure 1.21 - a. Les erreurs initiales qui sont introduites sur les poses notées $[\Delta_X, \Delta_Y, \Delta_Z, \Delta_{R_X}, \Delta_{R_Y}, \Delta_{R_Z}]$ sont $[2 \text{ m}, -2 \text{ m}, -2 \text{ m}, -5^\circ, -5^\circ, 15^\circ]$.

La caméra virtuelle converge vers la pose désirée comme le montre la différence entre l'image finale $\mathbf{I}(\mathbf{r}_f)$ et l'image désirée \mathbf{I}^* (Fig. 1.21 - c) et les erreurs de positions et d'orientations décroissent et tendent vers 0. L'erreur finale de l'asservissement visuel est $[4,62e-3 \text{ m}, -4,63e-4 \text{ m}, -3,59e-3 \text{ m}, -0,0006331^\circ, -0,0284^\circ, -0,001891^\circ]$, ce qui donne une norme de position de $6,27e-3 \text{ m}$. Nous pouvons remarquer que la précision diminue lorsque le nombre de DDL augmente.

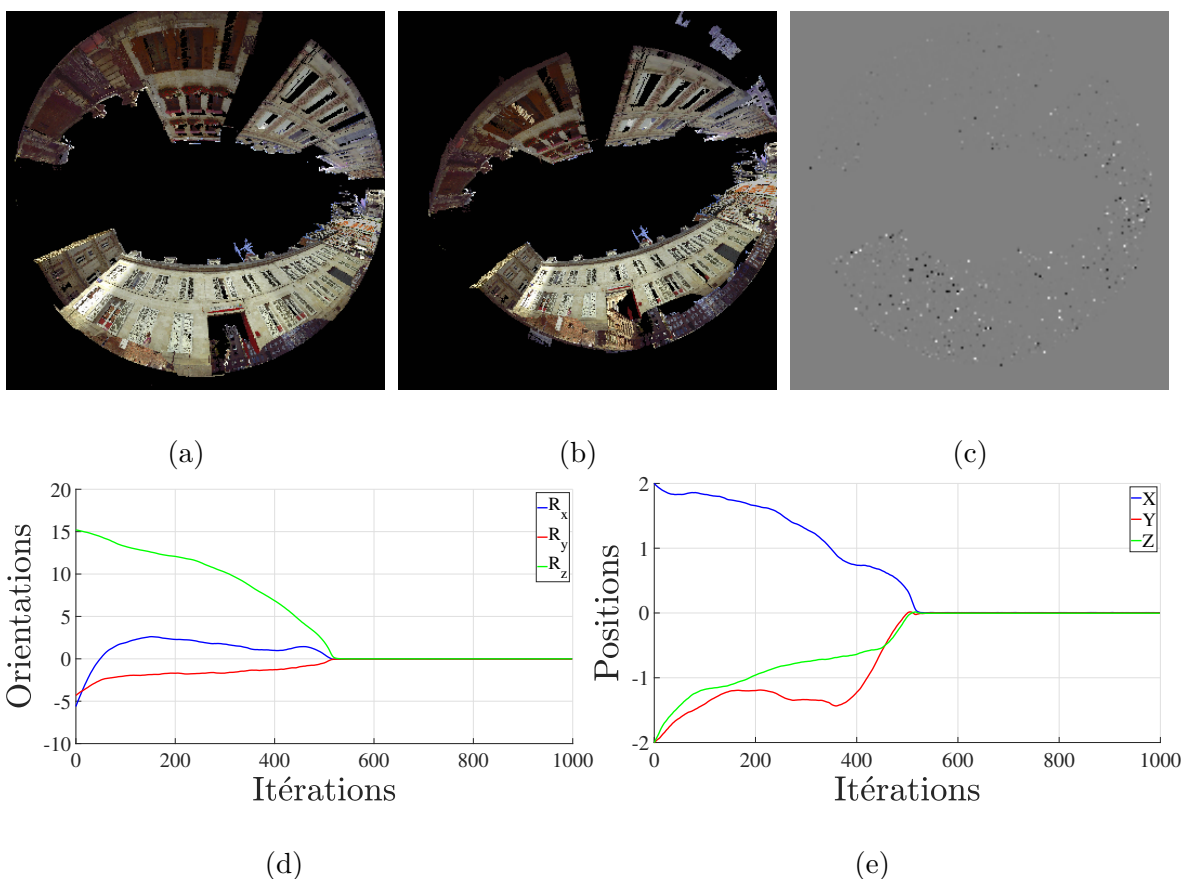


FIGURE 1.21 – Simulation en 6-DDL : erreur initiale : $[\Delta_X, \Delta_Y, \Delta_Z, \Delta_{R_X}, \Delta_{R_Y}, \Delta_{R_Z}] = [2 \text{ m}, -2 \text{ m}, -2 \text{ m}, -5^\circ, -5^\circ, 15^\circ]$, (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l'évolution de la position et (e) l'évolution de l'orientation de la caméra.

— **Simulation 2 :**

La pose désirée dans cet exemple (Fig. 1.22 - a) est la même que celle de l'expérience précédente. Les erreurs initiales de positionnement sont de $[10 \text{ m}, -4 \text{ m}, -2 \text{ m}, -10^\circ, -10^\circ, 15^\circ]$ et sont représentées par la figure 1.22 - b. La pose initiale de la caméra est donc plus éloignée de la pose désirée que dans l'exemple précédent.

Dans cet exemple, la caméra converge vers un minimum local et l'erreur finale est $[9.0571 \text{ m}, 0.7340 \text{ m}, -1.252 \text{ m}, 6.5710^\circ, -6.0326^\circ, 5.9280^\circ]$.

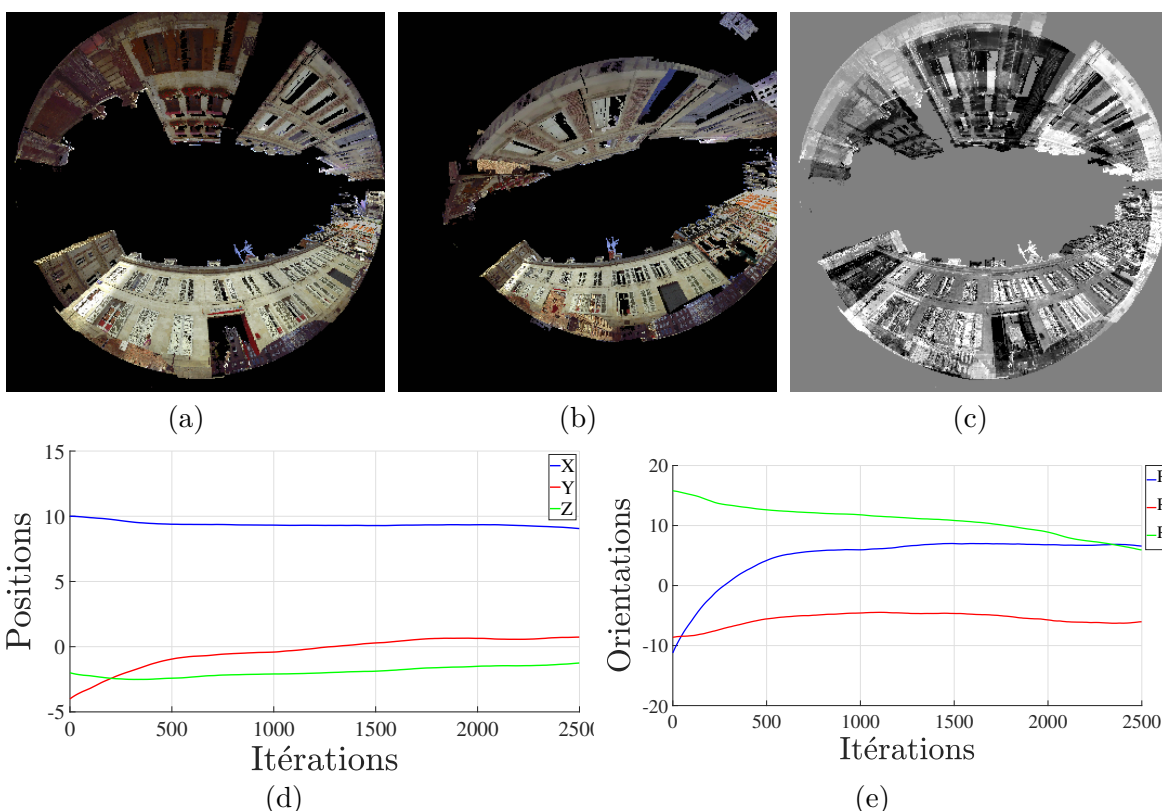


FIGURE 1.22 – Simulation en 6-DDL bis : erreur initiale : $[\Delta_X, \Delta_Y, \Delta_Z, \Delta_{R_X}, \Delta_{R_Y}, \Delta_{R_Z}] = [10 \text{ m}, -4 \text{ m}, -3 \text{ m}, -10^\circ, -10^\circ, 15^\circ]$, (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l'évolution de la position et (e) l'évolution de l'orientation de la caméra.

— **Simulation 3 :**

Nous prenons alors une autre scène où la pose désirée est montrée par la figure 1.23 - a. Les erreurs initiales de positionnement sont de $[-8 \text{ m}, -8 \text{ m}, -2 \text{ m}, -6^\circ, -13^\circ, 40^\circ]$. L'image initiale correspondante est donnée par la figure 1.23 - b.

Nous enregistrons en fin de simulation une convergence vers un minimum local : l'erreur finale de l'asservissement visuel est $[-17.02 \text{ m}, 1.71 \text{ m}, -49.12 \text{ m}, 31.84^\circ, 21.99^\circ, -33.56^\circ]$.

Les deux derniers exemples de simulation que nous venons de proposer s'inscrivent dans une démarche comparative pour les résultats du prochain chapitre de contributions.

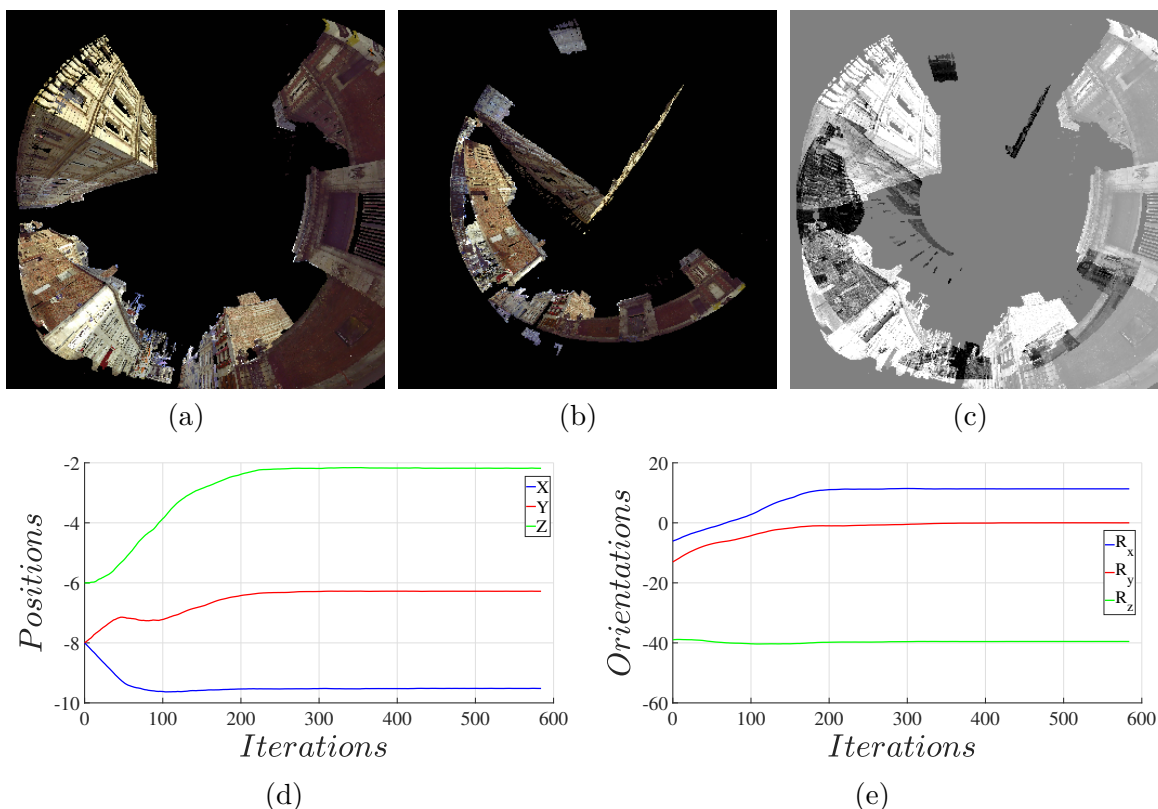


FIGURE 1.23 – Simulation en 6-DDL ter : erreur initiale : $[\Delta_X, \Delta_Y, \Delta_Z, \Delta_{R_X}, \Delta_{R_Y}, \Delta_{R_Z}] = [-8 \text{ m}, -8 \text{ m}, -6 \text{ m}, -6^\circ, -13^\circ, -38^\circ]$, (a) image désirée, (b) image initiale, (c) image de différence finale, (d) l'évolution de la position et (e) l'évolution de l'orientation de la caméra.

1.5 Conclusion

Dans ce premier chapitre, nous avons présenté les différents types de projections des caméras ainsi que l'asservissement visuel, en exploitant l'information photométrique.

L'asservissement visuel photométrique, utilisant l'intégralité de l'information visuelle, assure une grande précision de convergence et implique seulement un calcul du gradient de l'image.

Plusieurs simulations ont été réalisées pour illustrer le comportement et les limites de l'asservissement visuel photométrique avec différents degrés de liberté : domaine de convergence restreint lié à une fonction de coût contenant des minima locaux malgré l'utilisation des caméras omnidirectionnelles.

Dans la suite du manuscrit, nous nous attacherons à repousser les limites de l'asservissement visuel photométrique, en particulier son domaine de convergence, en proposant de représenter les images omnidirectionnelles par un mélange de gaussiennes photométriques en s'inspirant de travaux faits en vision perspective de Crombez *et al.* [CROMBEZ *et al.*, 2019].

Asservissement visuel omnidirectionnel basé mélanges de Gaussiennes photométriques

Sommaire

| | | |
|------------|---|-----------|
| 2.1 | Introduction | 36 |
| 2.2 | Modélisation d'une image optique | 36 |
| 2.3 | Représentation abstraite des pixels d'une image | 37 |
| 2.3.1 | Modèle de gaussienne photométrique | 37 |
| 2.3.2 | Mélange de gaussiennes photométriques | 40 |
| 2.4 | Mélanges de gaussiennes photométriques comme caractéristiques visuelles denses en asservissement visuel omnidirectionnel | 41 |
| 2.4.1 | Loi de commande | 42 |
| 2.4.2 | Impact du paramètre d'extension | 44 |
| 2.4.3 | Règle existante pour choisir les paramètres d'extension | 45 |
| 2.4.4 | Nouvelles règles pour choisir les paramètres d'extension | 46 |
| 2.5 | Évolution en simulation | 46 |
| 2.5.1 | Protocole | 46 |
| 2.5.2 | Ordres des grandeurs des MGP selon les 3 règles | 47 |
| 2.5.3 | Simulation en 1 DDL | 48 |
| 2.5.4 | Simulation en 2 DDL (2 translations) | 53 |
| 2.5.5 | Simulation en 3 DDL (3 translations) | 56 |
| 2.5.6 | Simulations en 6 DDL (3 translations et 3 rotations) | 59 |
| 2.5.7 | Évaluation multiple du domaine de convergence | 65 |
| 2.6 | Conclusion | 66 |

2.1 Introduction

L’asservissement d’une caméra vers une image cible revient à un problème d’optimisation non-linéaire. La solution optimale de ce problème dépend, entre autre, de la différence qualitative entre l’image courante et l’image désirée qui peut varier fortement en fonction de différents facteurs, par exemple la distance entre leurs deux poses. Ainsi, le recouvrement d’informations photométriques doit être important pour que l’asservissement puisse converger sans tomber dans un minimum local (voir Chapitre 1).

Dans ce chapitre, nous proposons une approche qui peut permettre de lisser suffisamment la fonction de coût afin de supprimer une partie des minima locaux.

Ceci peut être fait en changeant la représentation de l’information photométrique et lisser de manière réglable la fonction de coût et ainsi étendre le domaine de convergence. Ce chapitre s’appuie sur le Mélange de Gaussiennes Photométriques (MGP), proposé par [CROMBEZ et al., 2019] pour des images perspectives. Chaque pixel est représenté par une gaussienne pondérée par son intensité. La somme des gaussiennes de tous les pixels forme le MGP qui permet de convertir l’effet ponctuel et discret de l’information d’intensité en une information continue et définie partout.

Ce chapitre étend cette approche aux images omnidirectionnelles. Il présente comment l’utilisation de cette technique peut améliorer la convergence de l’asservissement visuel d’une caméra omnidirectionnelle. Ainsi, deux contributions sont apportées :

- Utilisation du MGP pour la réduction des minima locaux de la fonction de coût photométrique omnidirectionnelle.
- Détermination de la cohérence entre les paramètres d’extensions courant et désiré du MGP.

2.2 Modélisation d’une image optique

Lors d’une capture photographique, la lumière de la scène est quantifiée au moyen d’un capteur photosensible. Le rayonnement électromagnétique est converti en un signal électrique analogique, qui est ensuite amplifié et discrétisé par un convertisseur analogique-numérique. Une valeur d’intensité ou de couleur est associée à chaque pixel de l’image numérique obtenue, fonction de la quantité de particules de lumière ayant atteint le capteur.

Cette mesure de lumière peut être vue de façon abstraite comme une discrétisation spatiale et bidimensionnelle de l’image optique. La discrétisation spatiale d’un signal continu se réalise comme un échantillonnage en deux dimensions avec un pas régulier, qui peut se modéliser par une distribution de Dirac δ [DIRAC, 1929] :

$$\int_A \delta(x) dx = \begin{cases} 1 & \text{si } 0 \in A \\ 0 & \text{sinon} \end{cases} \quad (2.1)$$

avec $A \subseteq \mathbb{R}$.

La répétition régulière de cette distribution donne un peigne de Dirac \mathbf{III} , c'est-à-dire une série finie en deux dimensions séparées par un pas spatial $\mathbf{T} = (T_x, T_y)$. Ce pas doit respecter le théorème de Nyquist-Shannon [SHANNON, 1948], qui permet d'assurer que toute l'information contenue dans un signal analogique de bande de fréquence finie est bien conservée lors de l'échantillonnage en une séquence discrète. On définit $\mathbf{u} = (u, v)$ comme étant les coordonnées du signal discrétisé.

Le peigne de Dirac en deux dimensions se définit par :

$$\mathbf{III}_{\mathbf{T}}(x, y) = \sum_u \sum_v \delta(x - uT_x, y - vT_y) \quad (2.2)$$

avec x et $y \in \mathbb{R}$.

L'intensité lumineuse numérique $I_{\mathbf{u}}$ peut ainsi être définie par le produit entre l'image optique $g(x, y)$ et le peigne de Dirac sur chaque pixel¹ (représentation approximative Fig. 2.1). Ainsi chaque distribution de Dirac de coordonnées (uT_x, vT_y) prend l'intensité $I_{\mathbf{u}}$ [GONZÁLEZ et al., 1992] :

$$I_{\mathbf{u}} = \int_{\mathbb{R}} \int_{\mathbb{R}} g(x, y) \mathbf{III}_{\mathbf{T}}(x, y) dx dy = g(uT_x, vT_y) \quad (2.3)$$

2.3 Représentation abstraite des pixels d'une image

Nous cherchons par la suite à ajouter une couche d'abstraction à l'intensité afin de convertir cette valeur en une fonction continue qui puisse améliorer l'asservissement visuel.

La représentation de la valeur d'intensité d'un pixel par une fonction continue en deux dimensions peut permettre d'élargir le "potentiel d'attraction" de ce pixel lors de l'asservissement. Il paraît donc intéressant de remplacer la fonction de Dirac par une distribution étendue, comme une fonction gaussienne de valeur maximale au point $\mathbf{u} = (u, v)$ et tendant vers 0 à l'infini.

2.3.1 Modèle de gaussienne photométrique

Soient deux variables indépendantes u_g et v_g formant le vecteur $\mathbf{u}_g = (u_g, v_g)$, on peut définir une gaussienne 2D par la fonction de distribution suivante :

$$f(\mathbf{u}_g, \mathbf{u}, \Sigma) = A \exp\left(-\frac{1}{2}(\mathbf{u}_g - \mathbf{u})^T \Sigma^{-1}(\mathbf{u}_g - \mathbf{u})\right) \quad (2.4)$$

1. La fonction réponse de la caméra et la quantification sont ici ignorées pour simplifier.

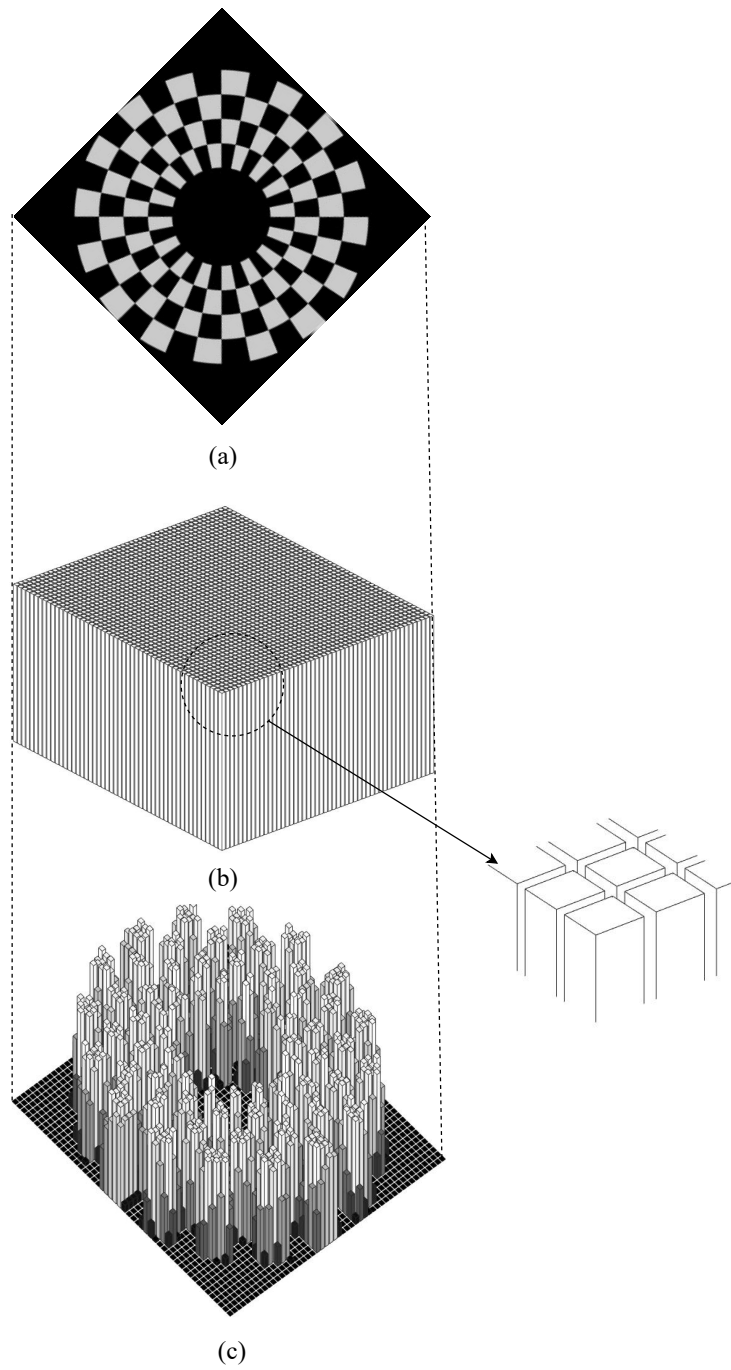


FIGURE 2.1 – Schéma représentant le produit du signal continu (a) avec un peigne (b), donnant lieu à une image numérique (c).

- avec A l'amplitude de la gaussienne, $\mathbf{u} = (u, v)$ l'espérance mathématique, Σ la matrice de covariance définie positive $\Sigma \in \mathbb{R}^{2 \times 2}$:

$$\Sigma = \begin{bmatrix} \sigma_u^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix} \quad (2.5)$$

- avec σ_u, σ_v les écarts-type de la fonction gaussienne.

Nous appelons $\mathbf{u} = (u, v)$ le centre de la gaussienne et (σ_u, σ_v) ses extensions selon, respectivement, les axes \vec{u}_g et \vec{v}_g . La figure 2.2 montre une surface gaussienne en deux dimensions d'amplitude $A = 2$, centrée en $(u, v) = (0, 0)$ avec $(\sigma_u, \sigma_v) = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$.

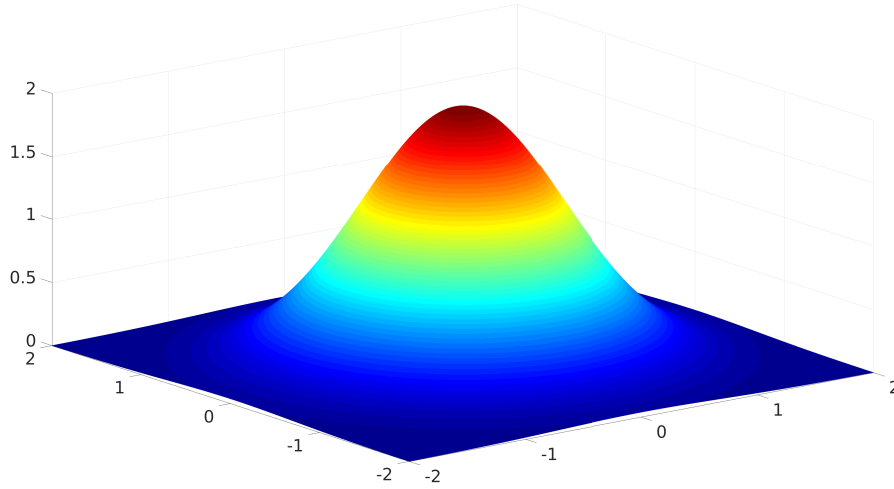


FIGURE 2.2 – Fonction gaussienne en deux dimensions centrée en $(u, v) = (0, 0)$.

Nous verrons par la suite qu'il est très intéressant de pouvoir agir sur l'extension des gaussiennes au cours de l'asservissement. En effet, le potentiel d'attraction attribué au pixel atteint son maximum à la position du pixel et est de plus en plus faible en fonction de la distance à ce centre. Nous n'étudierons que le cas de gaussiennes symétriques où σ_u et σ_v restent égaux à une variable λ_g qui nous servira de paramètre de réglage de l'extension des gaussiennes le long des axes \vec{u}_g et \vec{v}_g .

Ainsi, la représentation photométrique de la gaussienne en deux dimensions de l'équation 2.4, avec $I(\mathbf{u})$ représentant l'intensité du pixel \mathbf{u} , devient :

$$g(\mathbf{u}_g, \mathbf{u}, \Sigma) = I(\mathbf{u}) \exp\left(-\frac{1}{2}(\mathbf{u}_g - \mathbf{u})^T \Sigma^{-1}(\mathbf{u}_g - \mathbf{u})\right) \quad (2.6)$$

avec

$$\Sigma = \begin{bmatrix} \lambda_g^2 & 0 \\ 0 & \lambda_g^2 \end{bmatrix} \quad (2.7)$$

La gaussienne 2D est définie sur \mathbb{R}^2 à l'infini, mais on n'en considère que la portion incluse dans le cadre image par la suite.

2.3.2 Mélange de gaussiennes photométriques

Chaque pixel de l'image \mathbf{I} de taille $(P \times M)$ est modélisé par une gaussienne 2D exprimée par l'équation 2.6. Puis, nous combinons les $(P \times M)$ gaussiennes en les sommant selon \mathbf{u} , ce qui forme un mélange de gaussiennes [RASMUSSEN et al., 1999] :

$$G(\mathbf{I}, \mathbf{u}_g) = \sum_{\mathbf{u}} g(\mathbf{u}_g, \mathbf{u}, \Sigma) \quad (2.8)$$

La figure 2.3 représente le mélange de deux gaussiennes correspondant à deux pixels voisins. Pour un paramètre d'extension faible, le chevauchement peut être anecdotique. La figure 2.4 montre l'évolution du mélange de gaussiennes d'une image mono-dimensionnelle avec différentes valeurs du paramètre d'extension λ_g .

Nous observons sur la figure 2.4 - b que les mélanges de gaussiennes avec un paramètre d'extension faible (e.g. $\lambda_g = 0.1$) sont proches de l'image d'origine. Dans ce cas, l'asservissement basé mélange de gaussiennes photométriques à λ_g constant serait équivalent à l'asservissement visuel purement photométrique. Pour les autres valeurs de λ_g , les mélanges de gaussiennes sont de plus en plus lisses, ce qui augmente le potentiel d'attraction des différents pixels lors de l'asservissement visuel.

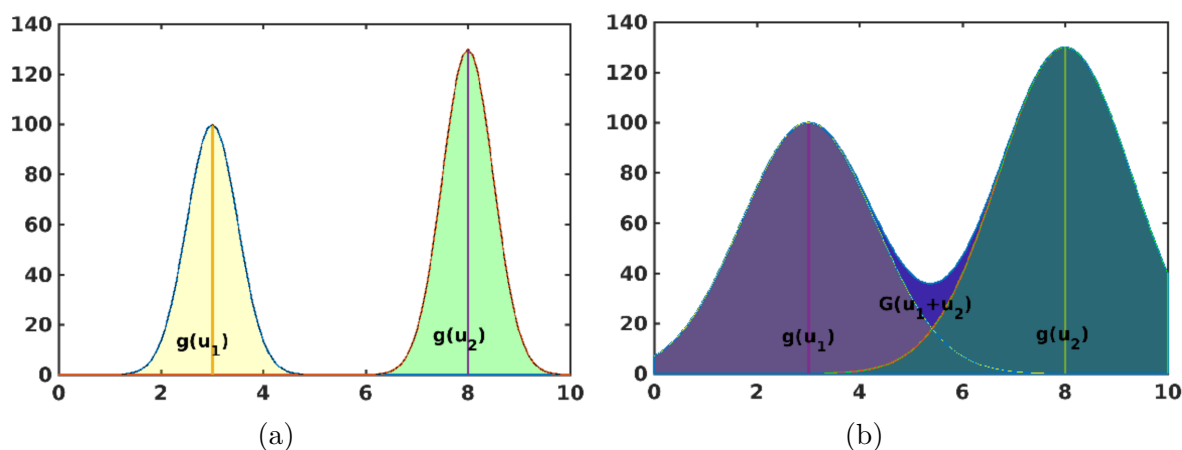


FIGURE 2.3 – Exemple d'un mélange de deux gaussiennes monodimensionnelles, calculées avec un paramètre d'extension faible (a) et avec un paramètre d'extension suffisant pour qu'il y ait chevauchement (b).

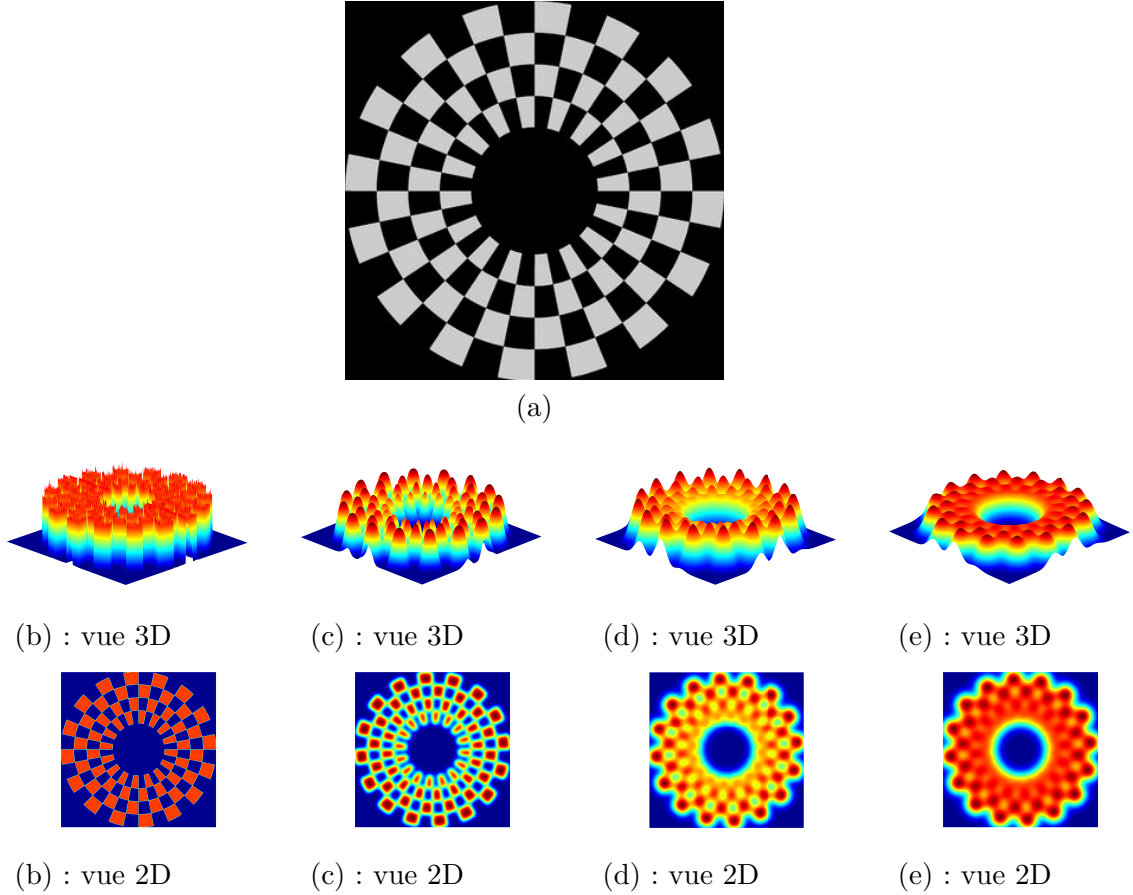


FIGURE 2.4 – Mélange de gaussiennes photométriques appliqué sur une image d’une mire omnidirectionnelle (a) avec différentes valeurs du paramètre d’extension : $\lambda_g = 0.1$ (b), $\lambda_g = 5$ (c), $\lambda_g = 10$ (d) et $\lambda_g = 12$ (e).

2.4 Mélanges de gaussiennes photométriques comme caractéristiques visuelles denses en asservissement visuel omnidirectionnel

Dans l’asservissement visuel omnidirectionnel purement photométrique, les intensités de l’image désirée \mathbf{I}^* acquise à la pose \mathbf{r}^* sont directement comparées avec celles de l’image courante \mathbf{I} acquise à la pose \mathbf{r} . Dans notre étude basée sur les mélanges de gaussiennes photométriques, l’objectif est de réguler le coût suivant :

$$\mathbf{e}(\mathbf{r}, \lambda_g) = G(\mathbf{I}(\mathbf{r}), \lambda_g) - G(\mathbf{I}^*, \lambda_g^*) \quad (2.9)$$

- $G(\mathbf{I}^*)$ est le mélange de gaussiennes calculé sur l’image omnidirectionnelle désirée avec un paramètre d’extension λ_g^* ,
- $G(\mathbf{I}(\mathbf{r}))$ est le mélange de gaussiennes calculé sur l’image omnidirectionnelle courante

à la pose \mathbf{r} avec un paramètre d'extension λ_g qui évolue durant l'asservissement.

2.4.1 Loi de commande

La pose optimale de la caméra et le paramètre d'extension optimal du MGP courant sont considérés comme étant la solution d'un problème d'optimisation non-linéaire. L'évolution des 6 DDL de la caméra ainsi que le paramètre d'extension sont assurés par la loi de commande de type Gauss-Newton :

$$\mathbf{v}_g = \mu \mathbf{L}_G^+ (G(\mathbf{I}^*) - G(\mathbf{I}(\mathbf{r}), \lambda_g)) \quad (2.10)$$

- $\mathbf{v}_g = (\mathbf{v}, \mathbf{w}, \dot{\lambda}_g)^T$, qui contient, dans l'ordre, les vitesses linéaires et angulaires de la caméra et la correction du paramètre d'extension,
- \mathbf{L}_G^+ est la pseudo-inverse de la matrice d'interaction associée au mélange de gaussiennes photométriques,
- μ est un paramètre qui règle la vitesse de convergence.

La matrice d'interaction \mathbf{L}_G est construite en empilant les jacobienes $\mathbf{L}_{G(\mathbf{u}_g)}$ correspondant à chaque échantillon du mélange de gaussiennes aux positions \mathbf{u}_g :

$$\mathbf{L}_G = \begin{bmatrix} \vdots & \vdots \\ \mathbf{L}_{G(\mathbf{u}_g)} & \lambda_{G(\mathbf{u}_g)} \\ \vdots & \vdots \end{bmatrix} \quad (2.11)$$

$\mathbf{L}_{G(\mathbf{u}_g)}$ est la sommation en \mathbf{u}_g de toutes les matrices d'interaction correspondant aux gaussiennes de chaque pixel :

$$\mathbf{L}_{G(\mathbf{u}_g)} = \sum_{\mathbf{u}} \mathbf{L}_{g(\mathbf{u}_g, \mathbf{u})} \quad (2.12)$$

$\mathbf{L}_{g(\mathbf{u}_g, \mathbf{u})}$ est la matrice d'interaction qui lie les variations de la gaussienne $g(\mathbf{u}_g, \mathbf{u})$ aux mouvements de la caméra.

La modélisation de la matrice d'interaction pour l'asservissement visuel purement photométrique suppose que l'intensité du pixel \mathbf{u} ne varie pas pour un mouvement $\delta \mathbf{u}$. Nous étendons cette supposition pour la gaussienne $g(\mathbf{u}_g, \mathbf{u})$ car la relation entre l'intensité du pixel et sa gaussienne est explicite :

$$g(\mathbf{u}_g + \delta \mathbf{u}_g, \mathbf{u}, t + \delta t) = g(\mathbf{u}_g, \mathbf{u}, t) \quad (2.13)$$

Nous appliquons un développement de Taylor d'ordre 1 de $g(\mathbf{u}_g + \delta \mathbf{u}_g, \mathbf{u}, t + \delta t)$ et nous obtenons :

$$g(\mathbf{u}_g + \delta\mathbf{u}_g, \mathbf{u}, t + \delta t) = g(\mathbf{u}_g, \mathbf{u}, t) + \frac{\partial g(\mathbf{u}_g, \mathbf{u}, t)}{\partial \mathbf{u}_g} \frac{d\mathbf{u}_g}{dt} + \frac{\partial g(\mathbf{u}_g, \mathbf{u}, t)}{\partial t} \quad (2.14)$$

Comme $g(\mathbf{u}_g + \delta\mathbf{u}_g, \mathbf{u}, t + \delta t) = g(\mathbf{u}_g, \mathbf{u}, t)$, on obtient l'équivalent de la contrainte du flot optique [HORN et al., 1981] (voir Section) :

$$\nabla_{\mathbf{u}_g} g^T \dot{\mathbf{u}}_g + \dot{g}(\mathbf{u}_g, \mathbf{u}, t) = 0 \quad (2.15)$$

avec $\nabla_{\mathbf{u}_g} g = (\nabla_{u_g} g, \nabla_{v_g} g)^T$ le gradient spatial de $g(\mathbf{u}_g, \mathbf{u}, t)$ et \dot{g} son gradient temporel.

La relation entre un point de mélange de gaussiennes et les vitesses de la caméra est assurée par l'expression suivante :

$$\dot{\mathbf{u}}_g = \mathbf{L}_{\mathbf{u}_g} \mathbf{v} \quad (2.16)$$

En introduisant les paramètres intrinsèques de la caméra, $\mathbf{L}_{\mathbf{u}_g}$ peut se décomposer comme étant :

$$\mathbf{L}_{\mathbf{u}_g} = \begin{bmatrix} \alpha_u & 0 \\ 0 & \alpha_v \end{bmatrix} \mathbf{L}_{\mathbf{x}} \quad (2.17)$$

avec $\mathbf{L}_{\mathbf{x}}$ (Eq. 1.21) la matrice d'interaction omnidirectionnelle associée au point exprimé dans le plan image normalisé.

En remplaçant l'équation 2.16 dans l'équation 2.15, nous obtenons :

$$\dot{g}(\mathbf{u}_g, \mathbf{u}, t) = -\nabla_{\mathbf{u}_g} g^T \mathbf{L}_{\mathbf{u}_g} \mathbf{v} \quad (2.18)$$

ce qui permet d'identifier l'expression de $\mathbf{L}_{g(\mathbf{u}_g, \mathbf{u})}$ 2.12 comme étant :

$$\mathbf{L}_{g(\mathbf{u}_g, \mathbf{u})} = -\nabla_{\mathbf{u}_g} g^T \mathbf{L}_{\mathbf{u}_g} \quad (2.19)$$

Pour compléter la matrice d'interaction (l'équation 2.19), il reste le gradient spatial à déterminer. Il peut être analytiquement calculé, en dérivant l'équation 2.6 selon \mathbf{u}_g [CROMBEZ et al., 2019] :

$$\begin{aligned} \nabla_{u_g} g &= \frac{\delta g(\mathbf{u}_g, \mathbf{u}, I_{\mathbf{u}}, \lambda_g)}{\delta u_g} \\ &= -\frac{I(\mathbf{u})(u_g - u)}{\lambda_g^2} \exp\left(-\frac{(u_g - u)^2 + (v_g - v)^2}{2\lambda_g^2}\right) \end{aligned} \quad (2.20)$$

Et

$$\begin{aligned}\nabla_{v_g} g &= \frac{\delta g(\mathbf{u}_g, \mathbf{u}, I_{\mathbf{u}}, \lambda_g)}{\delta v_g} \\ &= -\frac{I(\mathbf{u})(v_g - v)}{\lambda_g^2} \exp\left(-\frac{(u_g - u)^2 + (v_g - v)^2}{2\lambda_g^2}\right)\end{aligned}\quad (2.21)$$

Enfin, il reste à juxtaposer la variation du paramètre d'extension au fil de l'asservissement visuel dans l'équation 2.11. Pour cela, on calcule analytiquement la variation du mélange de gaussiennes par rapport au paramètre d'extension en dérivant l'équation 2.6 selon λ_g [CROMBEZ et al., 2019] :

$$\lambda_{G(\mathbf{u}_g)} = \frac{\delta g(\mathbf{u}_g, \mathbf{u}, I_{\mathbf{u}}, \lambda_g)}{\delta \lambda_g} = I(\mathbf{u}) \frac{((u_g - u)^2 + (v_g - v)^2)}{\lambda_g^3} \exp\left(-\frac{(u_g - u)^2 + (v_g - v)^2}{2\lambda_g^2}\right)\quad (2.22)$$

2.4.2 Impact du paramètre d'extension

Dans cette partie, nous proposons d'étudier l'impact de la représentation des images par mélange de gaussiennes photométriques sur la forme de la fonction de coût SSD. Nous les comparons ensuite avec la SSD photométrique. Afin de faciliter l'interprétation de cette comparaison, nous avons choisi une scène avec peu de détails et symétrique (Fig. 2.5 - a). Nous appliquons ensuite des translations sur la caméra virtuelle selon les axes \vec{X} et \vec{Y} autour de la pose désirée afin d'obtenir la fonction de coût. L'image désirée et les images courantes font une taille 200×200 pixels. Deux cas sont présentés ci-après. Dans le premier cas, on choisit le paramètre d'extension désiré $\lambda_g^* = \frac{\lambda_g}{2}$ et on fixe le paramètre d'extension λ_g [CROMBEZ et al., 2019]. Le paramètre λ_g^* permet de calculer le mélange de gaussiennes photométriques sur l'image désirée et le paramètre λ_g permet de calculer le mélange de gaussiennes photométriques sur les images obtenues durant les translations. Dans le second cas, nous prenons la même valeur pour le paramètre d'extension désiré et courant $\lambda_g^* = \lambda_g$.

Dans le cas de la fonction de coût purement photométrique, on constate que le domaine de convergence s'étend de -30 cm à 30 cm (Fig. 2.5 - b), en considérant la valeur 0 comme étant la pose optimale. Au delà de ces bornes, la fonction de coût est "plate".

Dans le cas des MGP et lorsque les deux paramètres λ_g et λ_g^* sont différents la fonction de coût est plus convexe que dans le cas purement photométrique (Fig. 2.5 - c). En effet le domaine de convergence varie entre -53 cm et 53 cm. Cette propriété montre que le mélange de gaussiennes photométriques étend le domaine de convergence. De plus, lorsque les deux paramètres sont identiques, la fonction de coût présente un minimum à zéro et son domaine de convergence s'étend de -68 cm à 68 cm (Fig. 2.5 - d).

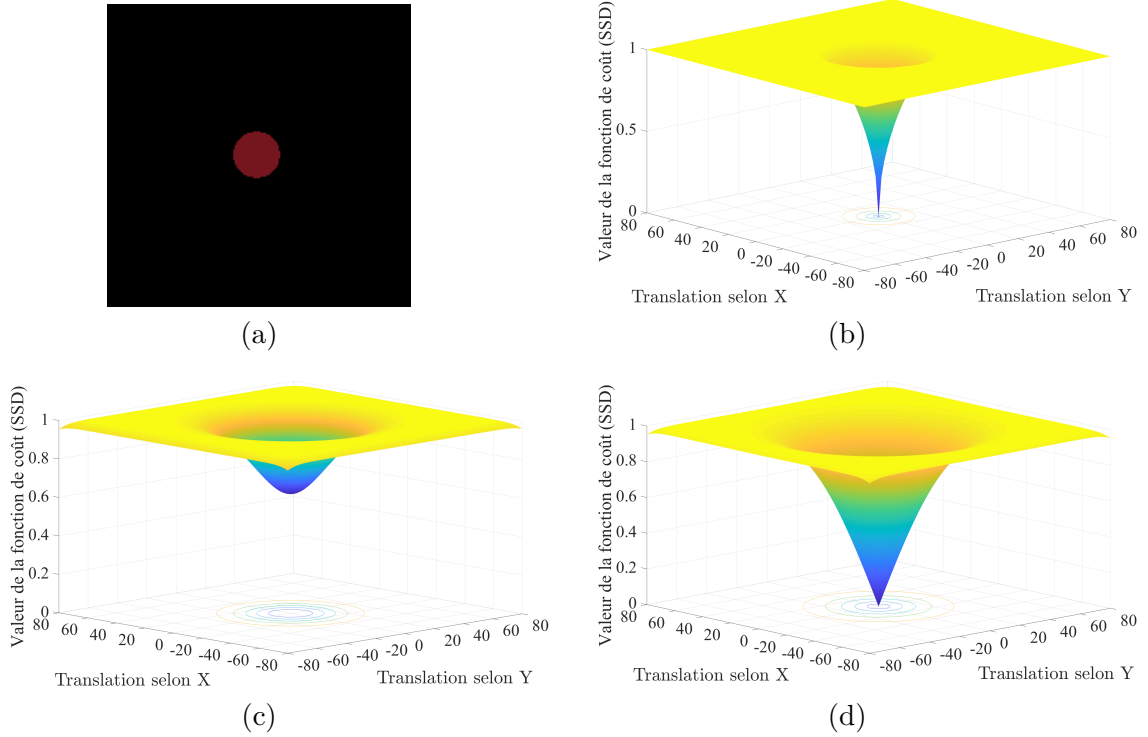


FIGURE 2.5 – Comparaison des fonctions de coût en 2-DDL : (a) image minimaliste, (b) fonction de coût basée sur la différence photométrique, (c) et (d) fonctions de coût à partir des images représentées avec le mélange de gaussiennes photométriques, pour (c) $\lambda_g^* = 7.5$ et $\lambda_g = 15$, pour (d) $\lambda_g^* = \lambda_g = 15$.

2.4.3 Règle existante pour choisir les paramètres d’extension

De manière générale, le mélange de gaussiennes photométriques élargit le domaine de convergence de la fonction de coût. Cependant, le choix du paramètre d’extension joue un rôle important. Dans [CROMBEZ et al., 2019], le paramètre d’extension λ_g des mélanges de gaussiennes photométriques est optimisé dans la loi de commande (Eq. 2.10) pour améliorer la convergence et la précision de l’asservissement visuel basé MGP. Pour atteindre ce comportement idéal, [CROMBEZ et al., 2019] procède en deux étapes durant l’asservissement visuel :

- Étape 1, avec λ_g grand (la valeur exacte dépend des expériences) et $\lambda_g^* = \frac{\lambda_g}{2}$, à l’initialisation ;
- Étape 2, avec la constante $\lambda_g = \lambda_g^* = 1$.

L’étape 1 assure un grand domaine de convergence et l’étape 2 assure la précision. Nous nommons ces deux étapes la *Règle0*.

2.4.4 Nouvelles règles pour choisir les paramètres d'extension

La *Règle0* qui pose $\lambda_g = 2\lambda_g^*$ à l'initialisation de l'étape 1 peut conduire à une divergence inattendue. Ce réglage peut en fait conduire à des MGP courants et désirés d'ordres de grandeur très différents (c'est pourquoi le coût minimum n'est pas nul en figure 2.5 - c). Pour résoudre ce problème, nous avons proposé et étudié deux nouvelles règles :

- *Règle1* : Étape 1, $\lambda_g = \lambda_g^*$ grand avec λ_g contrôlé; Étape 2 avec la constante $\lambda_g = \lambda_g^* = 1$
- *Règle2* : Étape 1, $\lambda_g = \lambda_g^*$ grand avec λ_g contrôlé; Étape 2 avec $\lambda_g = \lambda_g^* = 1$ à l'initialisation, en continuant de contrôler λ_g .

La différence entre ces deux dernières règles réside dans l'étape 2. Le choix de contrôler le paramètre d'extension λ_g ou pas va nous permettre de savoir quel est l'impact sur la précision et quelle règle donnera une précision proche du photométrique.

2.5 Évolution en simulation

2.5.1 Protocole

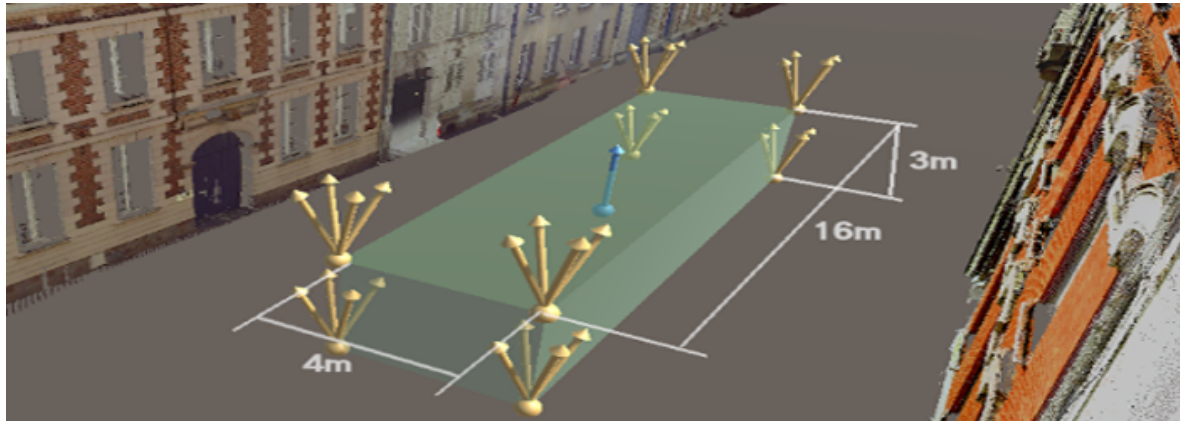
Dans un premier temps (sections 2.5.2 à 2.5.6), nous allons suivre le protocole déjà présenté dans la section 1.4.1, où nous avons asservi la vitesse en simulation sur un seul DDL. Ensuite, des DDL supplémentaires ont été ajoutés pour rendre la simulation plus réaliste. L'asservissement visuel basé MGP est appliqué avec les trois règles (*Règle0*, *Règle1* et *Règle2*). Les MGP désirés et courants sont calculés à partir du rendu de l'image omnidirectionnelle en environnement virtuel. Dans une démarche comparative, nous avons repris les mêmes poses désirées que dans le chapitre 1, mais avec des poses initiales plus éloignées de leur pose désirée.

Dans un second temps (2.5.7), et afin d'évaluer le domaine de convergence de façon plus systématique, 64 poses initiales \mathbf{r} sont générées autour de diverses poses souhaitées \mathbf{r}^* avec des combinaisons de transformations $t_X = \{-8 \text{ m}, 8 \text{ m}\}$, $t_Y = \{-2 \text{ m}, 2 \text{ m}\}$, $t_Z = \{-1.5 \text{ m}, 1.5 \text{ m}\}$, $\theta w_X = \{-10^\circ, 10^\circ\}$, $\theta w_Y = \{-10^\circ, 10^\circ\}$ et $\theta w_Z = \{-15^\circ, 15^\circ\}$. Les positions initiales forment ensemble un volume inclus dans des rues de l'environnement virtuel d'une largeur d'environ 12 m (Fig. 2.6 - a). Les 7 poses désirées \mathbf{r}^* sont réparties dans le modèle 3D ((Fig. 2.6 - b)-(Fig. 2.6 - h)).

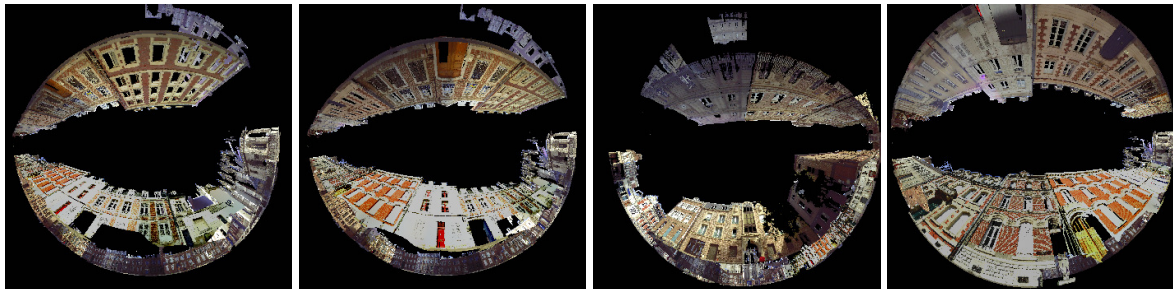
Pour l'asservissement visuel basé MGP, nous limitons l'étape 1 à 120 itérations. Pour les trois règles, nous avons choisi $\lambda_g = 15$ à l'initialisation afin d'avoir une zone de recouvrement suffisante entre \mathbf{G}^2 et \mathbf{G}^*^3 . Le gain de contrôle μ dans la loi de commande (Eq. 2.10) est fixé à 0,2 pour chaque asservissement visuel.

2. $G(\mathbf{I}(\mathbf{r}))$ est remplacé par \mathbf{G} pour faciliter la lecture.

3. $G(\mathbf{I}^*)$ est remplacé par \mathbf{G}^* pour faciliter la lecture.



(a)



(b)

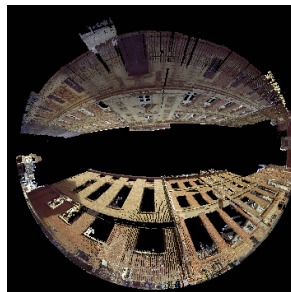
(c)

(d)

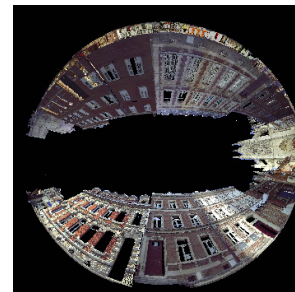
(e)



(f)



(g)

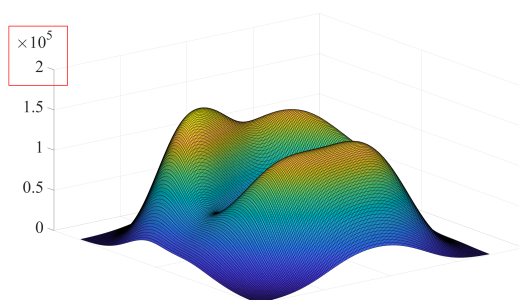


(h)

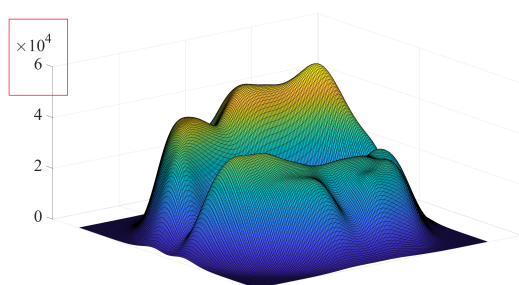
FIGURE 2.6 – Évaluation du domaine de convergence : (a) Visualisation d'un parallélépipède englobant les 64 positions initiales de la caméra (flèches dorées) autour d'une position souhaitée (flèche bleue).; (b-h) Images rendues aux sept poses souhaitées.

2.5.2 Ordres des grandeurs des MGP selon les 3 règles

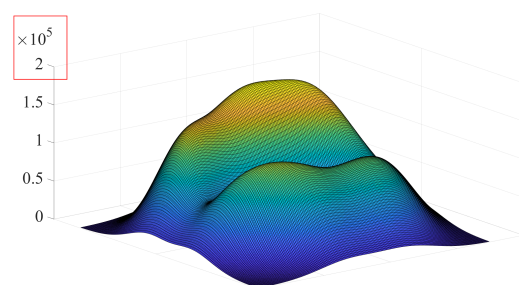
L'initialisation avec la *Règle0* peut engendrer une différence d'amplitude très importante entre le MGP courant (Fig. 2.7 - a) et le MGP désiré (Fig. 2.7 - b), comme évoqué en 2.4.4, alors qu'avec la *Règle1* ce n'est pas le cas (Fig. 2.7 - c).



(a) MGP initial



(b) MGP désiré *Règle0*



(c) MGP désiré *Règle1*

FIGURE 2.7 – Amplitude des MGP : (a) MGP initial, (b) MGP désiré *Règle0*, (c) MGP désiré *Règle1*.

2.5.3 Simulation en 1 DDL

Nous rappelons que dans l'exemple déjà présenté dans le chapitre 1 (Figure 1.15), le domaine de convergence photométrique autour de la pose désirée était limité entre -2 m et $+2$ m. Dans cet exemple, nous appliquons une translation pure d'une valeur de 20 m selon l'axe \vec{X} depuis la même pose désirée. La pose initiale se trouve donc en dehors du domaine de convergence photométrique. La figure 2.8 - a représente le rendu de la caméra virtuelle à partir de la pose désirée \mathbf{r}^* et la figure 2.8 - b son rendu à la pose initiale \mathbf{r}_0 .

Nous calculons ensuite le \mathbf{G}^* (Fig. 2.8 - d) sur l'image désirée $\mathbf{I}(\mathbf{r}^*)$ et le $\mathbf{G}(\mathbf{r}_0)$ (Fig. 2.8 - e) sur l'image initiale $\mathbf{I}(\mathbf{r}_0)$ selon l'étape 1 de la *Règle0*.

Nous pouvons observer que l'erreur entre les mélanges de gaussiennes photométriques désirés et courants, calculée à partir de l'image désirée, et les différentes images obtenues via la caméra virtuelle au cours de l'asservissement visuel, décroît au fil des itérations et tend vers 0 (Fig. 2.8 - g). Le paramètre d'extension λ_g , durant l'étape 1, décroît pour converger vers λ_g^* .

Afin de tester les limites du domaine de convergence en utilisant la *Règle0* pour la même

pose désirée (Fig. 2.8 - a), nous effectuons une translation pure selon l'axe \vec{X} d'une valeur de 30 m (Fig. 2.9 - b). Dans ce cas, le processus d'optimisation converge vers un minimum local (Fig. 2.9 - c).

Ensuite, nous prenons l'exemple précédent avec une translation de 30 m puis nous appliquons la *Règle1*. Nous pouvons observer (Fig. 2.10) que la caméra converge vers la pose désirée avec cette initialisation et que l'erreur entre le MGP désiré et le MGP courant augmente au début en raison de la diminution de la valeur de λ_g durant son optimisation par rapport à λ_g^* , cette diminution créant une petite différence entre le MGP courant et désiré. L'erreur décroît ensuite au fil des itérations et tend vers 0 (Fig. 2.10 - g). Le paramètre d'extension λ_g dans un premier temps décroît, puis remonte vers le paramètre d'extension désiré λ_g^* . L'étape 2 de la *Règle1* laisse une erreur résiduelle (Fig. 2.10 - c) d'une valeur finale Δ_X de $3,3e - 2$ m.

La *Règle2* est appliquée au même exemple que celui présenté précédemment et le comportement est identique pour l'étape 1. Pour l'étape 2, une amélioration est constatée au niveau de la précision, avec une erreur finale Δ_X (Fig. 2.11 - c) de l'ordre de $4,0e - 6$ m. Cette précision submillimétrique est proche du cas photométrique ($1,5e - 8$ m) comme vue dans la section 1.4.2.1 figure 1.15.

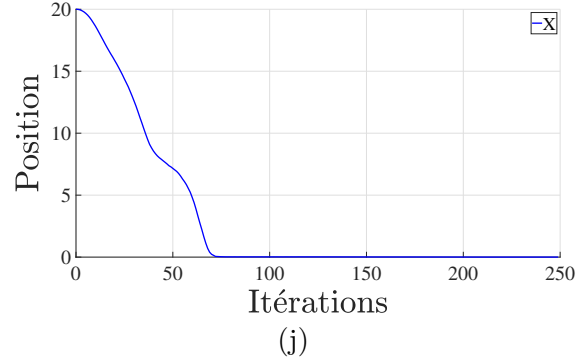
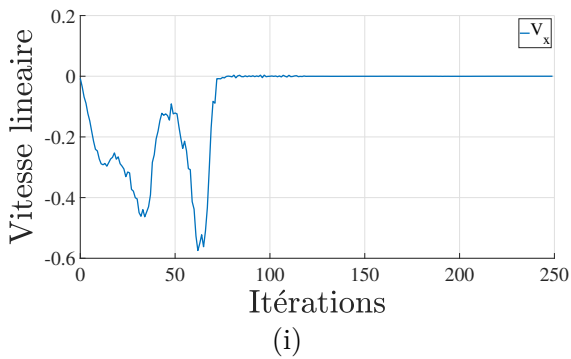
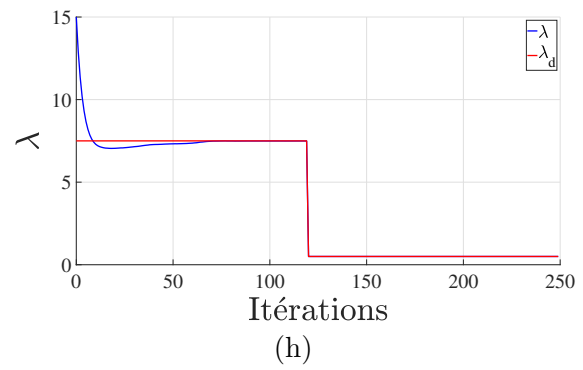
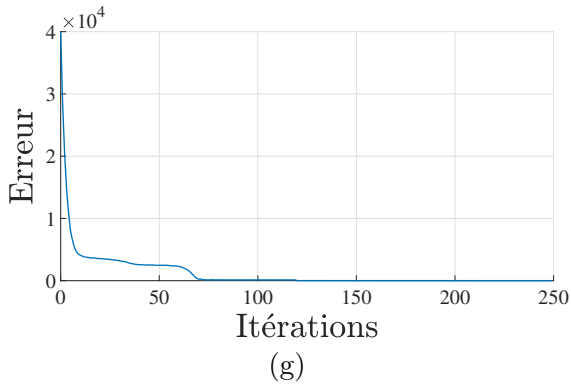
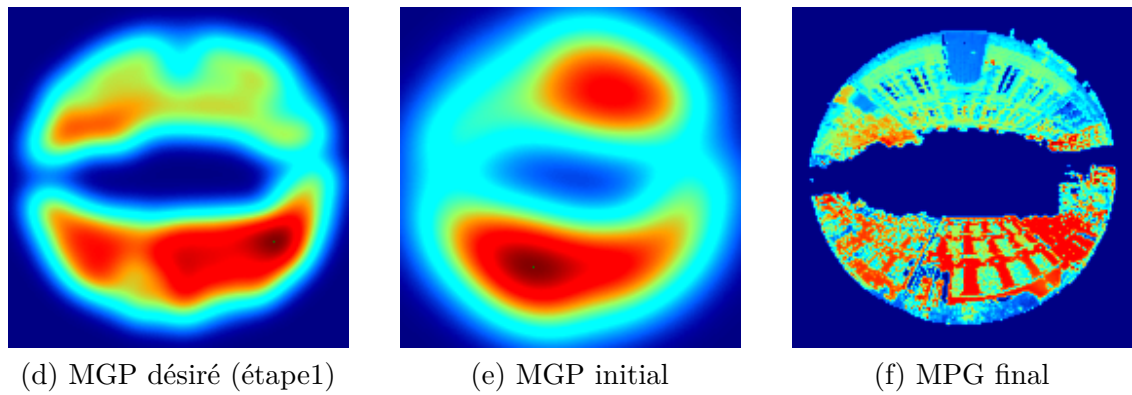


FIGURE 2.8 – Simulation en 1-DDL règle 0 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution de la vitesse et (j) Évolution de la position de la caméra virtuelle.

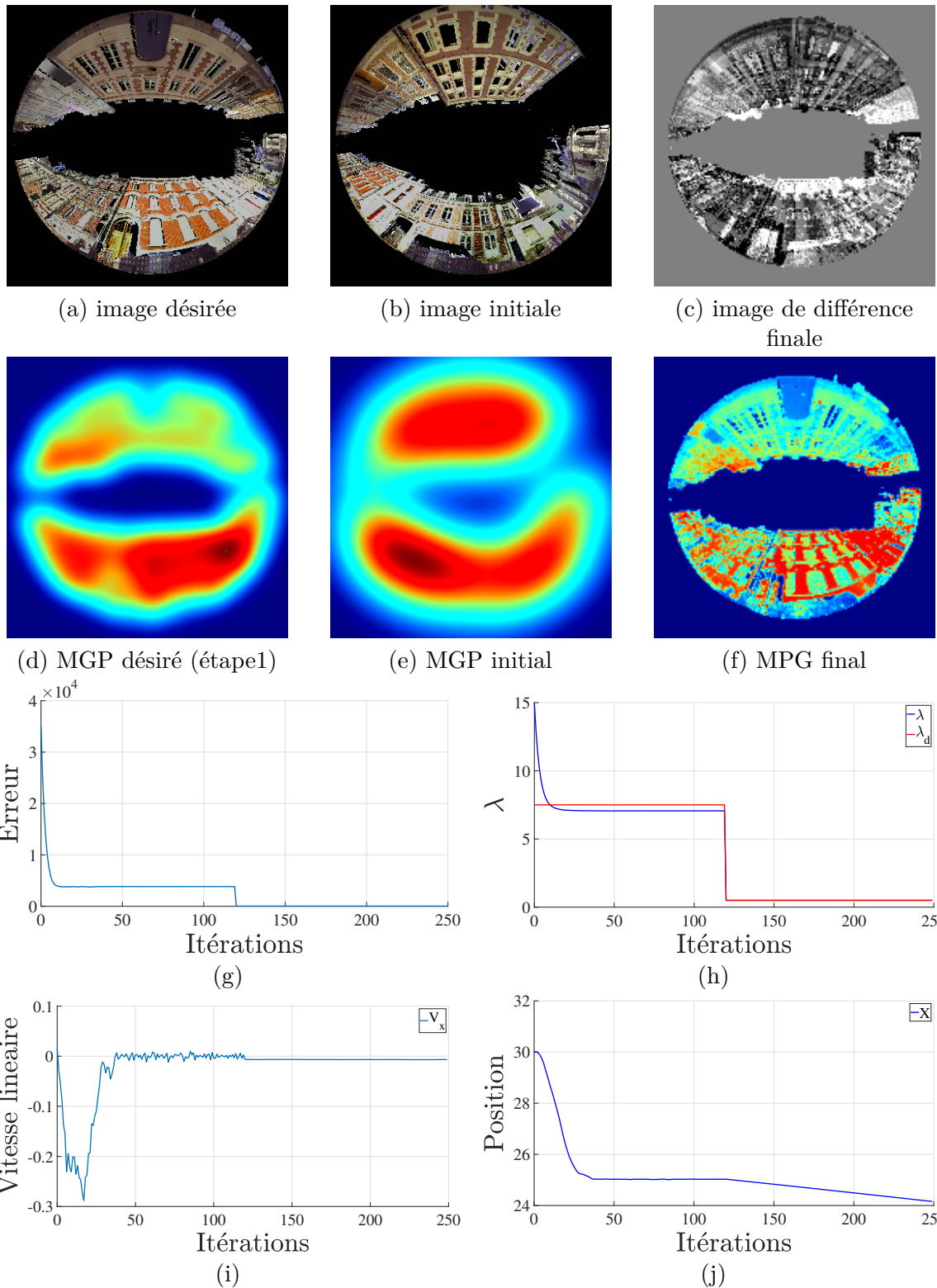


FIGURE 2.9 – Simulation en 1-DDL règle 0 bis : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution de la vitesse et (j) Évolution de la position de la caméra virtuelle.

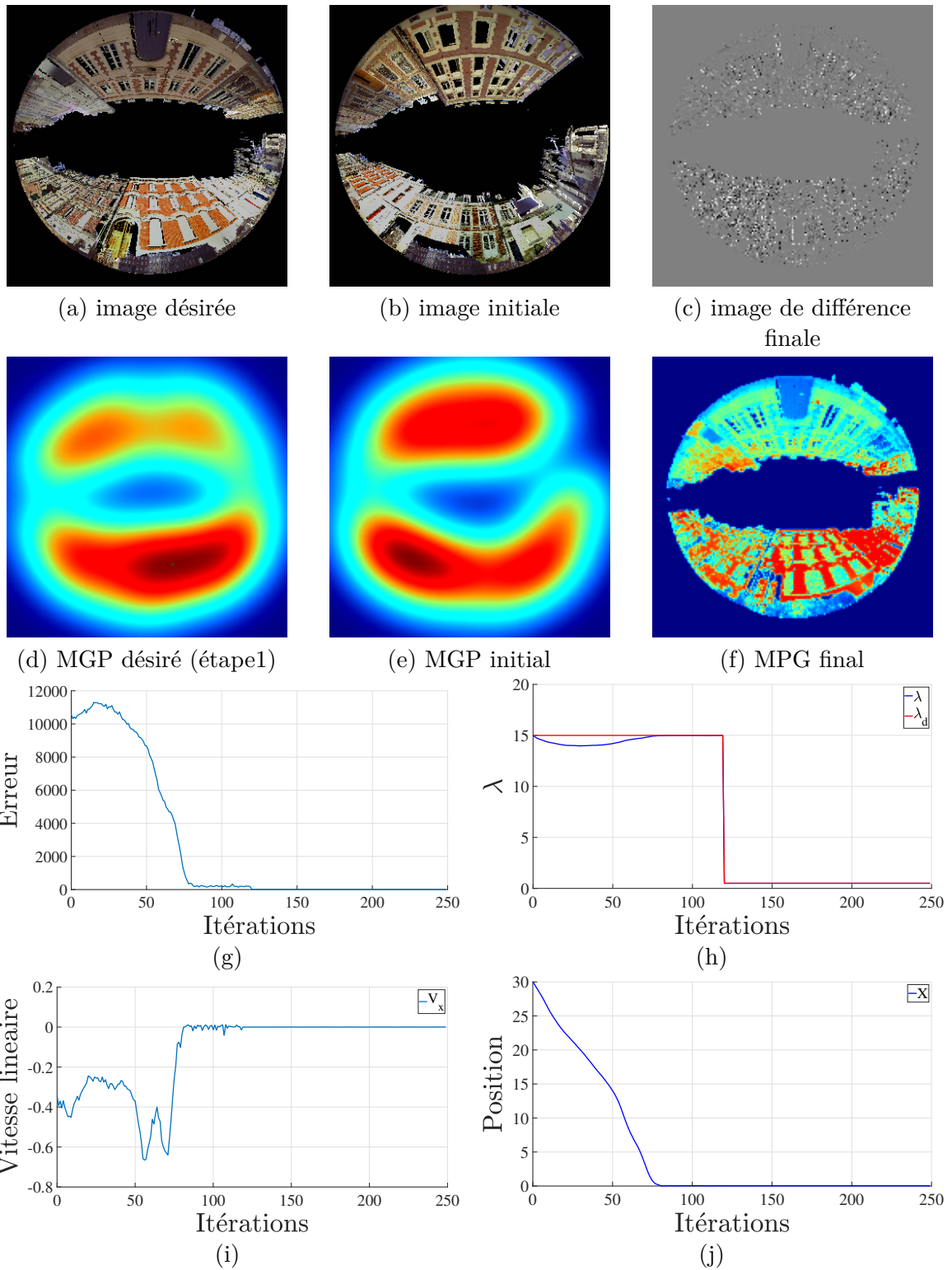


FIGURE 2.10 – Simulation en 1-DDL règle 1 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution de la vitesse et (j) Évolution de la position de la caméra virtuelle.

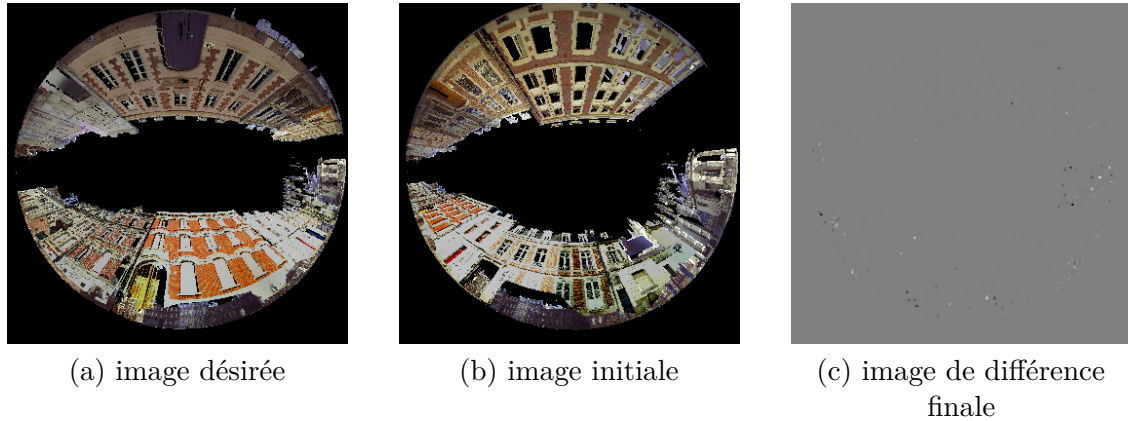


FIGURE 2.11 – Simulation en 1-DDL règle 2

2.5.4 Simulation en 2 DDL (2 translations)

Afin d'élargir notre champ d'étude, nous ajoutons un DDL au processus d'optimisation.

Dans cette section, nous allons appliquer des translations en dehors du domaine de convergence photométrique. Les résultats des asservissements visuels basés MGP selon les initialisations *Règle0*, *Règle1*, et *Règle2* seront comparés entre eux ainsi qu'avec l'asservissement visuel purement photométrique.

La pose initiale est obtenue en faisant une translation de -10 m selon l'axe \vec{X} et -4 m selon l'axe \vec{Y} . La figure 2.12 - b représente l'image initiale **I**, obtenue par la caméra virtuelle située à la pose initiale \mathbf{r}_0 . Le MGP désiré (Fig. 2.12 - d) et le MGP initial (Fig. 2.12 - e) sont calculés selon la *Règle0*.

Après 150 itérations (Fig. 2.12 - c), la caméra virtuelle converge vers un minimum local (Fig. 2.12 - j). Dans ce cas, la *Règle0* conduit au même résultat que l'asservissement visuel purement photométrique.

En appliquant la *Règle1*, nous constatons que l'erreur décroît au fil des itérations (Fig. 2.13 - g). L'erreur finale (Fig. 2.13 - c) $[\Delta_X, \Delta_Y]$ est de l'ordre de $[0.071 \text{ m}, 0.0572 \text{ m}]$ au bout de 250 itérations.

Dans l'exemple précédent, l'application de la *Règle1* entraîne une erreur résiduelle d'environ 9 cm . Cette erreur est élevée pour un asservissement visuel dense. La *Règle2* résout ce problème comme on peut le voir sur la figure (Fig. 2.14 - c). L'erreur finale $[\Delta_X, \Delta_Y]$ est de l'ordre de $[5,5e - 6 \text{ m}, 9,07e - 5 \text{ m}]$ avec une norme de position de l'ordre de $9,09e - 5 \text{ m}$. Nous pouvons remarquer la même précision submillimétrique photométrique dans la section 1.4.2.2.

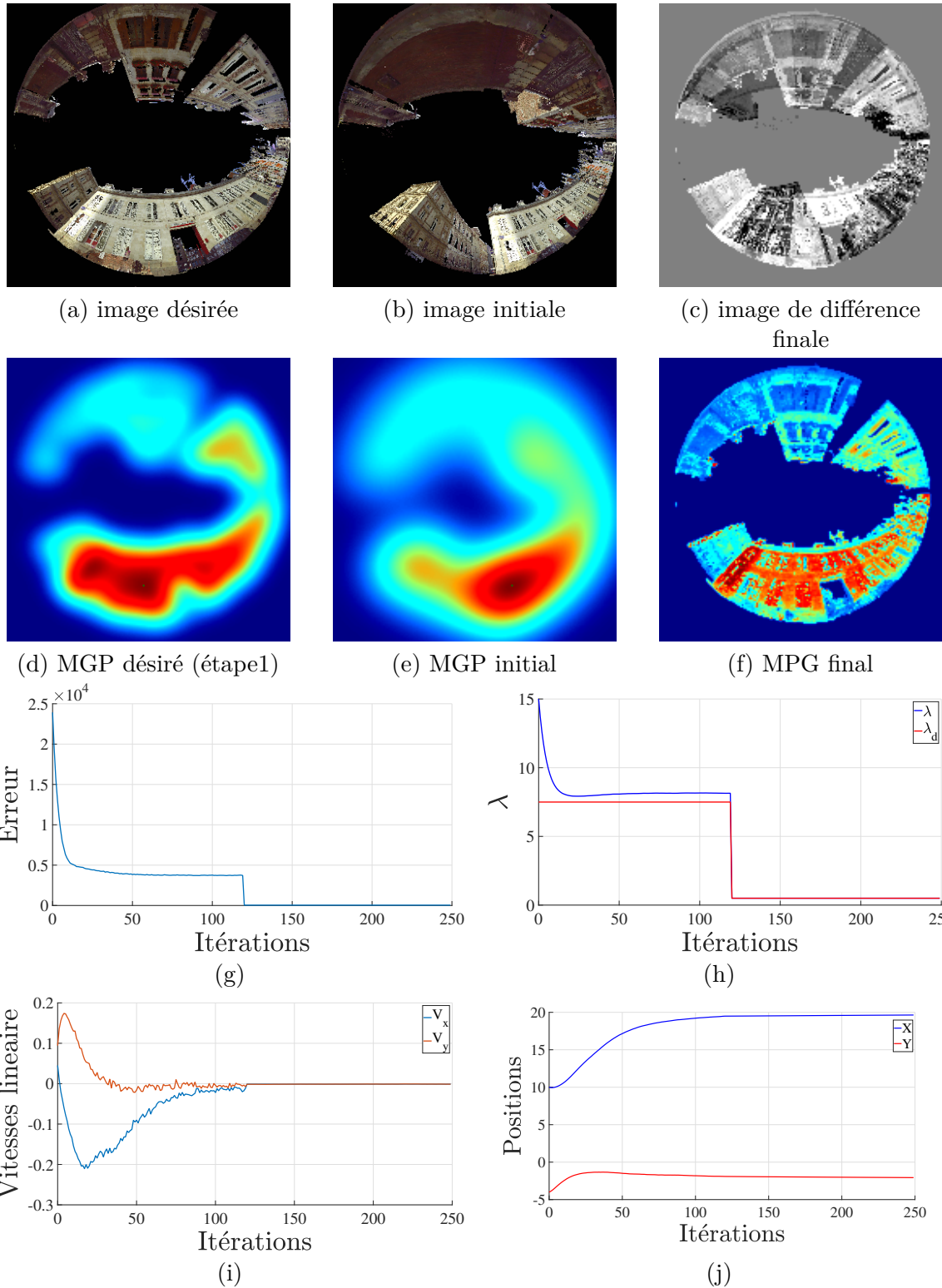


FIGURE 2.12 – Simulation en 2-DDL règle 0 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution des vitesses linéaires et (j) Évolution de la position de la caméra virtuelle.

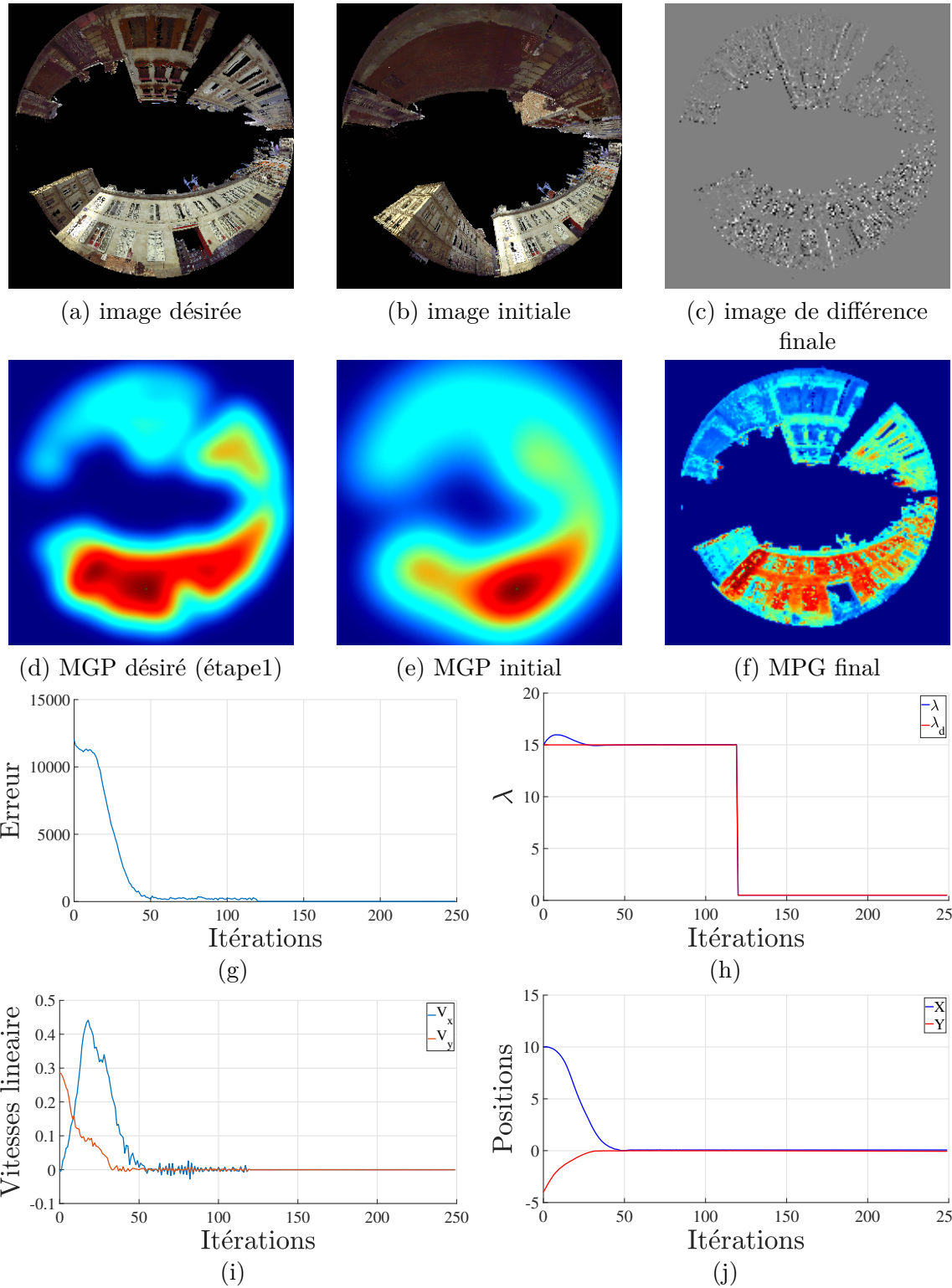


FIGURE 2.13 – Simulation en 2-DDL règle 1 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution des vitesses linéaires et (j) Évolution de la position de la caméra virtuelle.

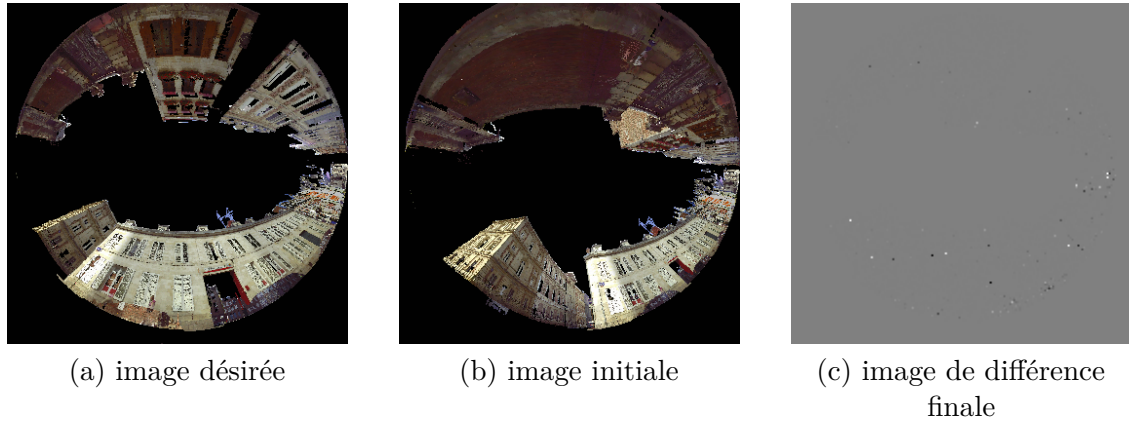


FIGURE 2.14 – Simulation en 2-DDL règle 2

2.5.5 Simulation en 3 DDL (3 translations)

Dans cette simulation, le nombre de DDL est augmenté en ajoutant la translation selon \vec{Z} . La figure 2.15 - b représente l'image initiale $\mathbf{I}(\mathbf{r}_0)$ de l'asservissement. La pose initiale \mathbf{r}_0 de la caméra virtuelle est obtenue en appliquant une première translation d'une valeur de -3 m selon l'axe \vec{Z} , une deuxième de -15 m selon l'axe \vec{X} et une troisième de -2 m selon l'axe \vec{Y} . L'éloignement de la pose désirée est ainsi plus important que dans l'exemple 1.20 où la caméra a convergé vers un minimum local.

En appliquant la *Règle0* la caméra virtuelle converge vers un minimum local (Fig. 2.15 - c), proche du minimum global. En effet, l'erreur finale de positionnement $[\Delta_X, \Delta_Y, \Delta_Z]$ est de l'ordre de $[1,38e - 2 \text{ m}, 8,14e - 2 \text{ m}, 0,12 \text{ m}]$, avec une norme de $0,14$ m.

L'application de la *Règle1* engendre une convergence de la caméra virtuelle vers un autre minimum local. L'erreur finale selon les trois dimensions est de l'ordre de $[1,2e - 3 \text{ m}, 4,9e - 4 \text{ m}, 3,0e - 4 \text{ m}]$ et sa norme est de $1,45e - 3$ m. L'erreur est moindre qu'avec l'initialisation de la *Règle1* mais toujours importante (Fig. 2.16 - c).

Nous constatons en appliquant la *Règle2* que la caméra virtuelle converge vers la pose désirée 2.17. À l'étape 1, l'erreur entre le MGP désiré et le MGP courant diminue au début puis augmente en raison de la diminution de la valeur de λ_g en regard de λ_g^* , diminution créant une différence entre \mathbf{G} et \mathbf{G}^* , puis décroît au fil des itérations en convergeant vers 0 (Fig. 2.17 - g). L'erreur finale est de l'ordre de $[6,6e - 4 \text{ m}, 3,2e - 4 \text{ m}, 8,2e - 05 \text{ m}]$, soit une norme de position de $7,4e - 4$ m, en accord avec la précision submillimétrique des approches photométriques (Fig. 2.17 - c).

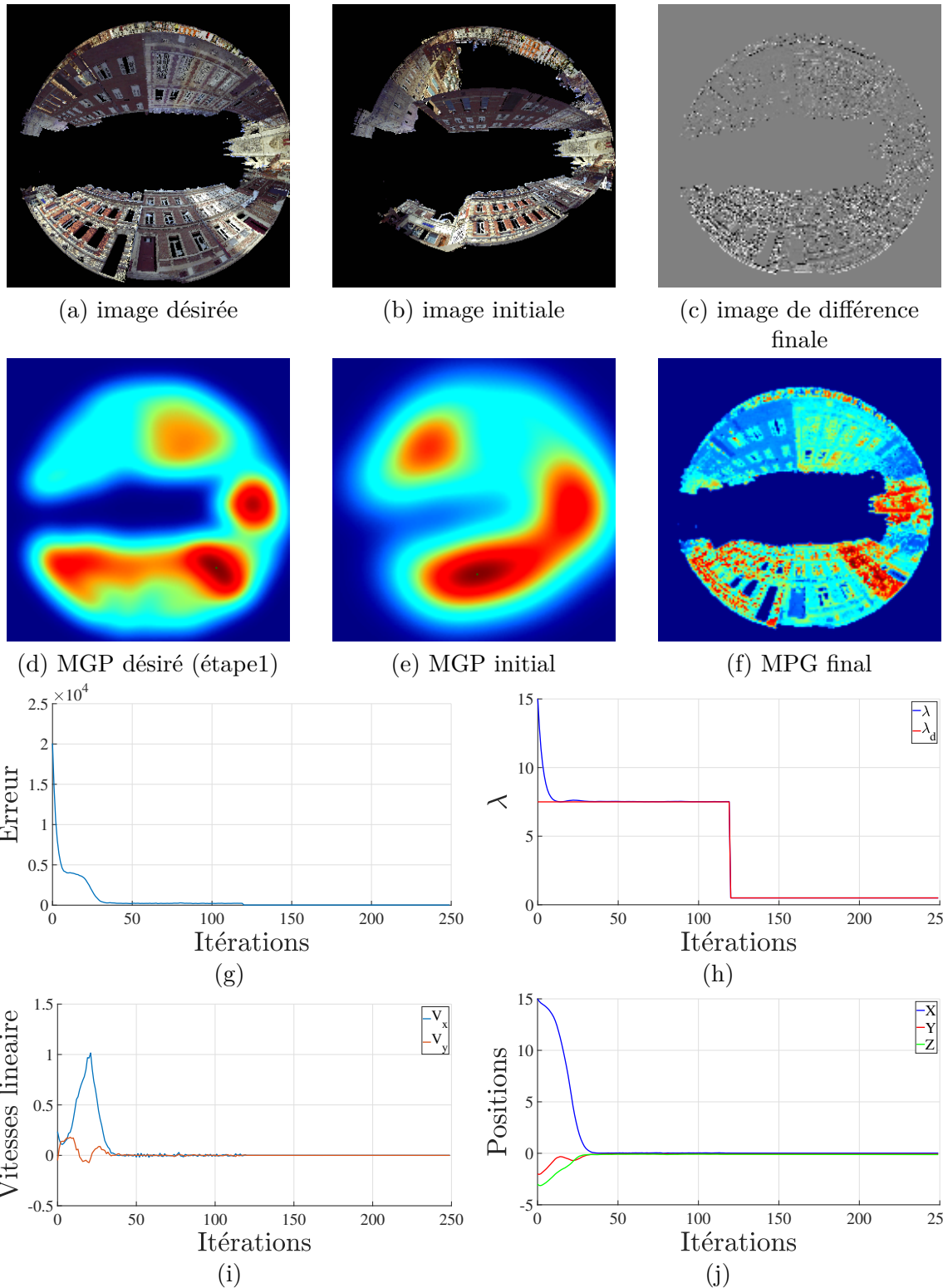


FIGURE 2.15 – Simulation en 3-DDL règle 0 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution des vitesses linéaires et (j) Évolution de la position de la caméra virtuelle.

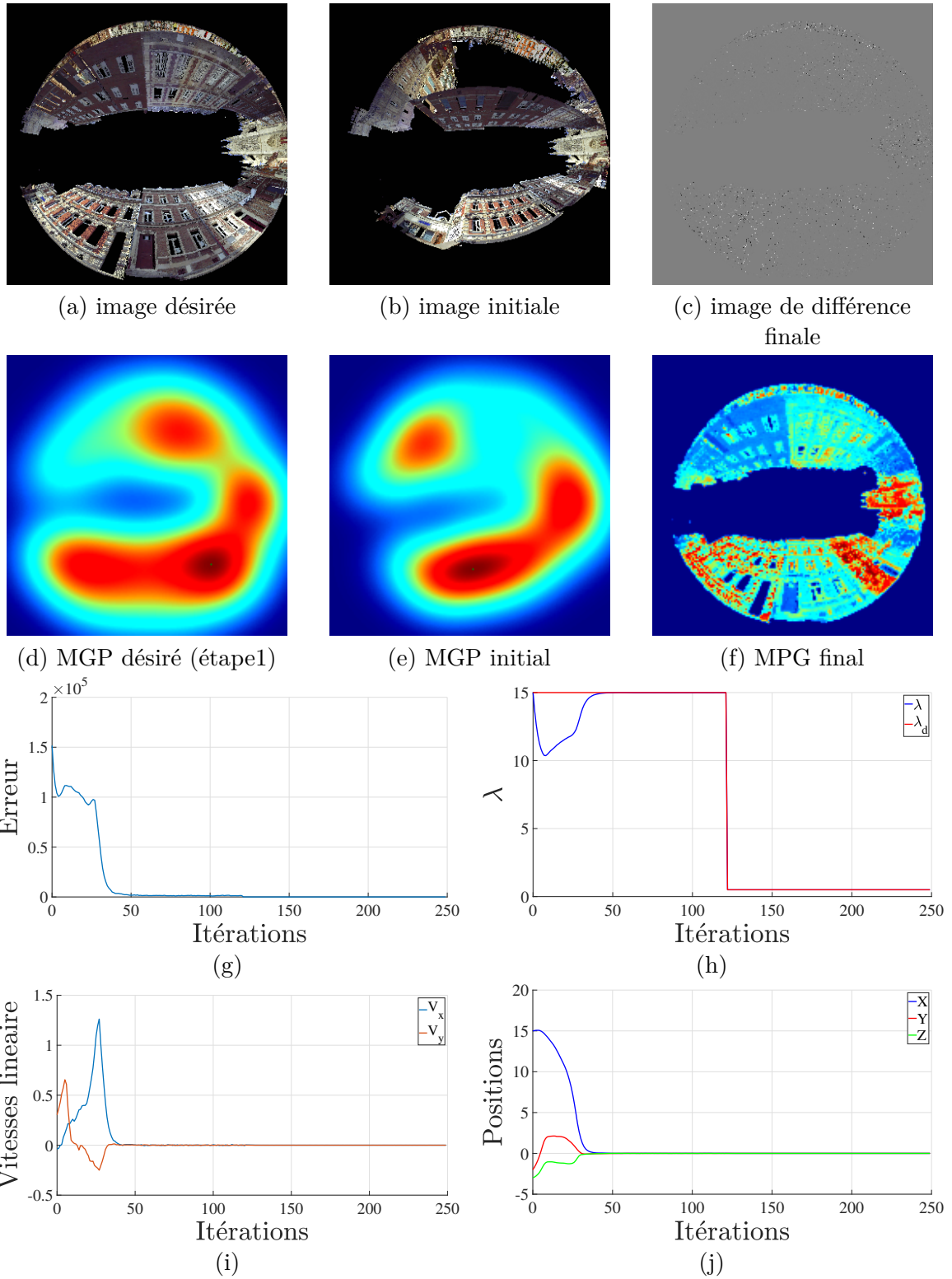


FIGURE 2.16 – Simulation en 3-DDL règle 1 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution des vitesses linéaires et (j) Évolution de la position de la caméra virtuelle.

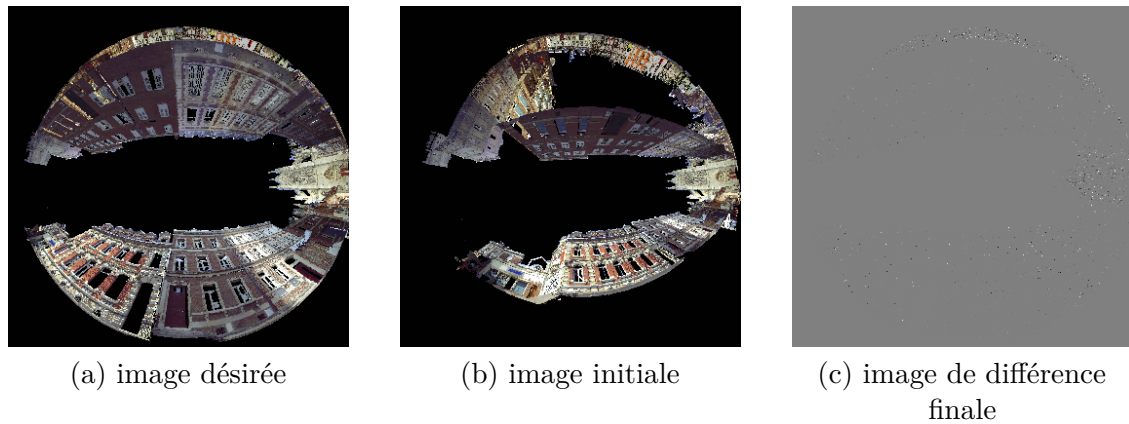


FIGURE 2.17 – Simulation en 3-DDL règle 2.

2.5.6 Simulations en 6 DDL (3 translations et 3 rotations)

2.5.6.1 Exemple 1 :

Pour cette expérience (Fig. 2.18), tous les DDL sont activés. L'erreur initiale sur les 6 DDL $[\Delta_X, \Delta_Y, \Delta_Z, \Delta_{R_X}, \Delta_{R_Y}, \Delta_{R_Z}]$ est de $[10 \text{ m}, -4 \text{ m}, -3 \text{ m}, -10^\circ, -10^\circ, 15^\circ]$. La pose initiale de la caméra est la même que dans l'exemple 1.4.2.4 en terme d'éloignement de la pose désirée où la caméra virtuelle a convergé vers un minimum local.

Nous remarquons dans ce cas et avec la *Règle0* que la caméra converge à un minimum local (Fig. 2.18 - c) avec une erreur finale de l'ordre de $[0,04 \text{ m}, 0,032 \text{ m}, 0,0012 \text{ m}, 0,014^\circ, 0,037^\circ, 0,0402^\circ]$, ce qui donne une norme de position de $0,0521 \text{ m}$.

L'application de la *Règle1* conduit la caméra virtuelle vers un autre minimum local avec une erreur "finale" de l'ordre de $[0,015 \text{ m}, 0,0012 \text{ m}, 0,011 \text{ m}, 0,5^\circ, 0,37^\circ, 0,05^\circ]$, ce qui donne une norme de $0,019 \text{ m}$.

Nous prenons maintenant le même exemple mais en appliquant la deuxième étape de la *Règle2*. La caméra virtuelle converge vers la pose désirée avec une erreur finale de l'ordre de $[3,5e-4 \text{ m}, 4,4e-4 \text{ m}, 2,372e-5 \text{ m}, 0,0013^\circ, 0,00024^\circ, 0,00201^\circ]$ et une norme de $5,7e-4 \text{ m}$ encore plus précise qu'avec la *Règle1*.

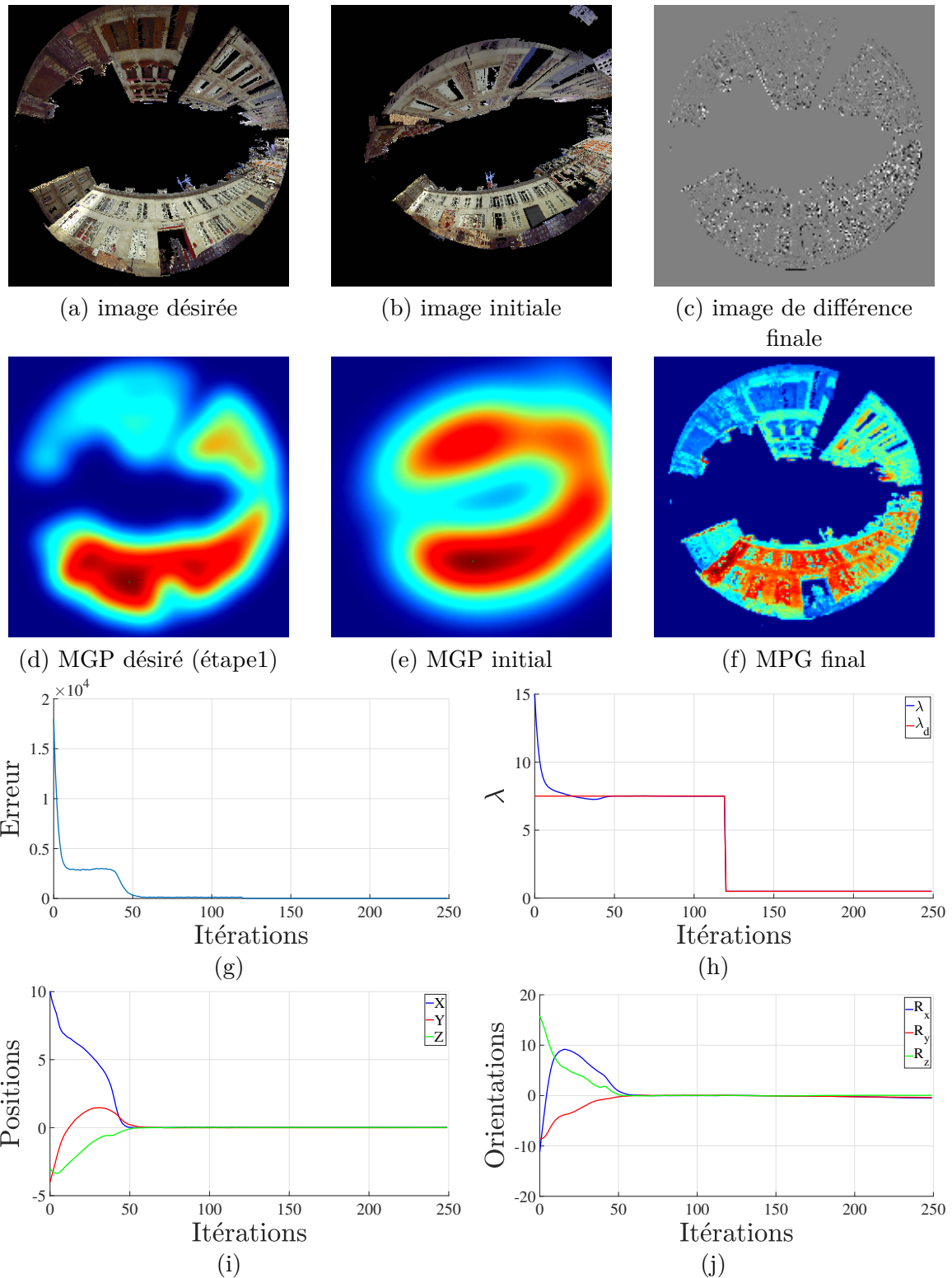


FIGURE 2.18 – Simulation en 6-DDL règle 0 : (g) Évolution de l’erreur, (h) Évolution du paramètre d’extension λ_g , (i) Évolution de la position de la caméra et (j) Évolution de l’orientation de la caméra virtuelle.

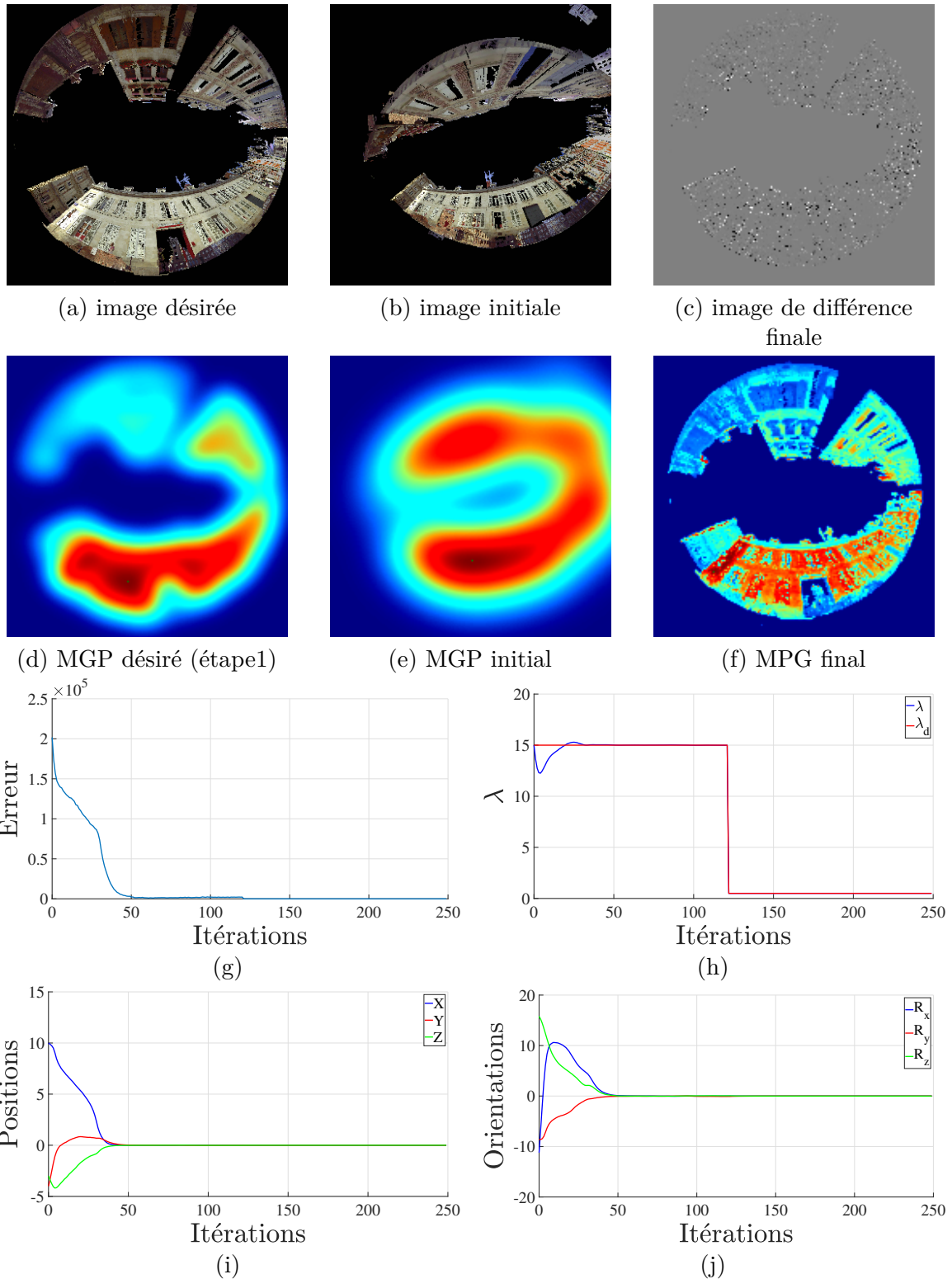


FIGURE 2.19 – Simulation en 6-DDL règle 1 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution de la position de la caméra et (j) Évolution de l'orientation de la caméra virtuelle.

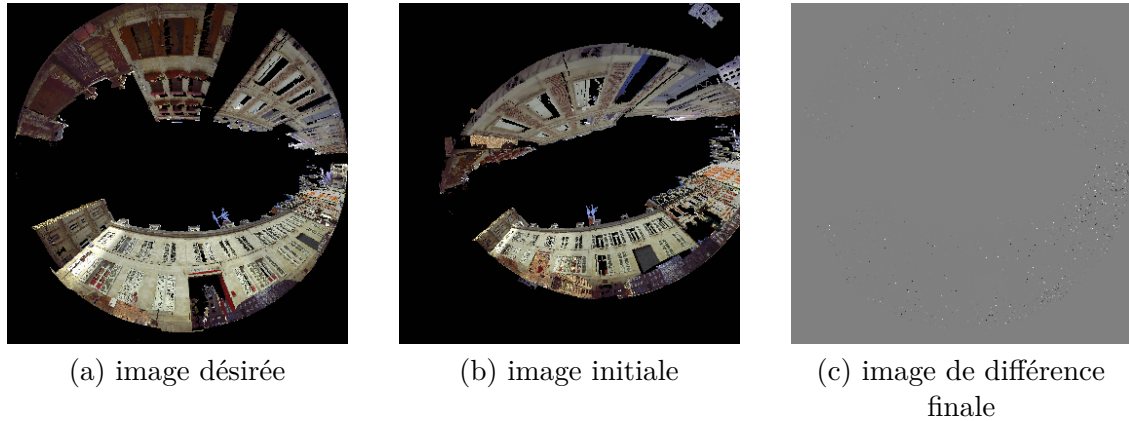


FIGURE 2.20 – Simulation en 6-DDL règle 2

2.5.6.2 Exemple 2 :

L'erreur initiale est de $[-8 \text{ m}, -8 \text{ m}, -6 \text{ m}, -6^\circ, -13^\circ, -38^\circ]$ (Fig. 2.21). La pose initiale de la caméra virtuelle est la même que dans l'exemple 1.4.2.4 en terme d'éloignement de la pose désirée où la caméra virtuelle diverge.

L'initialisation selon la *Règle0* ne permet pas la convergence vers la pose désirée. Or, selon la *Règle1*, nous observons que l'erreur entre le MGP désiré et le MGP courant décroît au début puis croît en raison de la valeur de λ_g en regard de λ_g^* , cette augmentation créant une différence entre \mathbf{G} et \mathbf{G}^* , puis décroît à nouveau au fil des itérations (Fig. 2.22 - g). La caméra virtuelle tend vers un minimum local à proximité de la pose désirée avec une erreur finale de l'ordre de $[0.0498 \text{ m}, 0.0776 \text{ m}, 0.0777 \text{ m}, 0.589^\circ, 0.465^\circ, 0.046^\circ]$ et une norme de 0.12 m .

L'initialisation selon la *Règle2* permet à la caméra virtuelle de converger vers la pose désirée avec une erreur finale de l'ordre de $[0.0010 \text{ m}, 0.0001 \text{ m}, 0.0004 \text{ m}, 0.0002^\circ, 0.0049^\circ, 0.0006^\circ]$, ce qui donne une norme finale de 0.0011 m , moindre qu'avec les règles précédentes.

Malgré une erreur initiale très importante entre la pose initiale et la pose désirée dans les 6DDL, l'asservissement visuel basé MGP converge vers la pose désirée comme le montre la figure 2.23 - c.

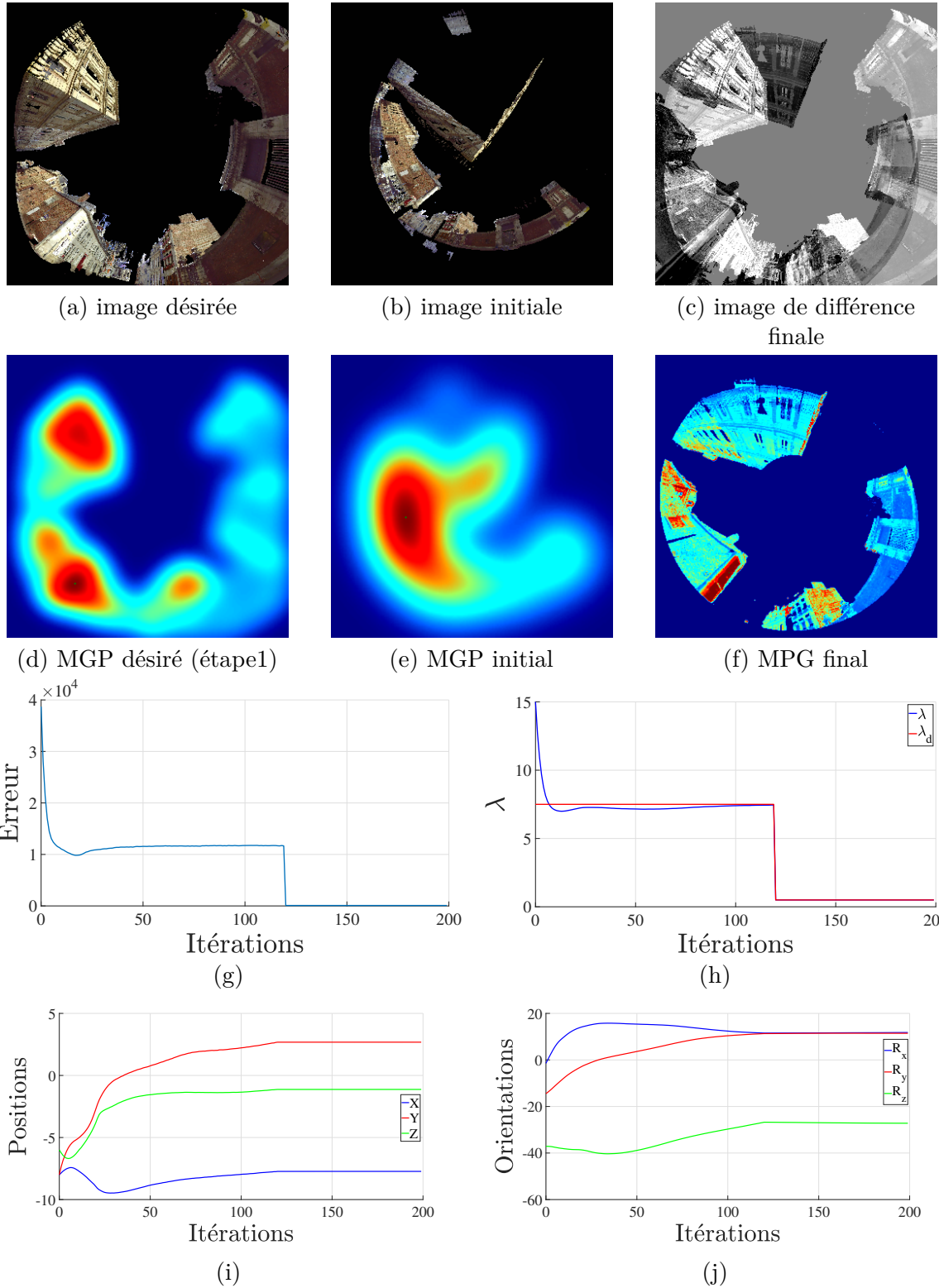


FIGURE 2.21 – Simulation en 6-DDL bis règle 0 : (g) Évolution de l'erreur, (h) Évolution du paramètre d'extension λ_g , (i) Évolution de la position de la caméra et (j) Évolution de l'orientation de la caméra virtuelle.

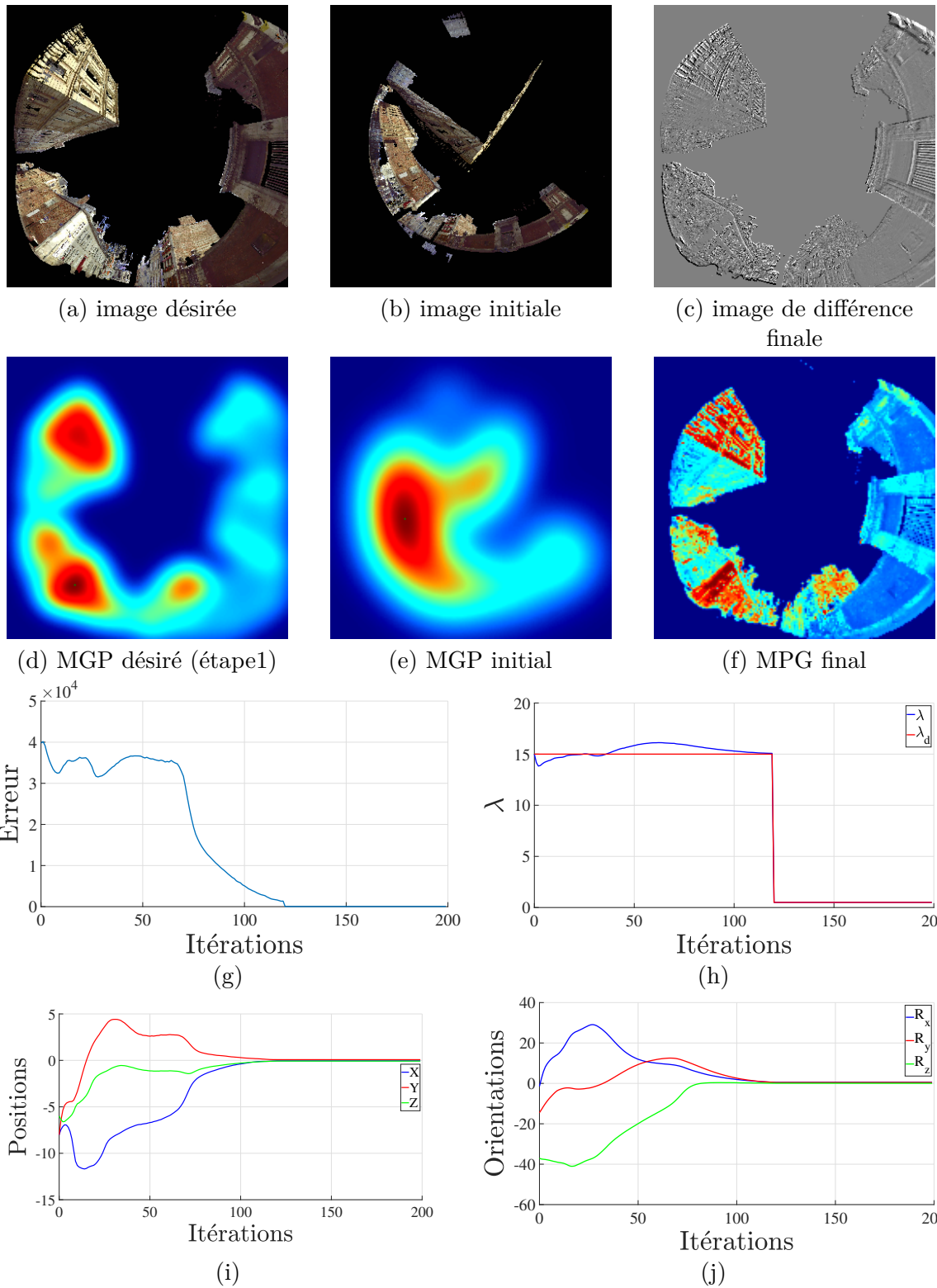


FIGURE 2.22 – Simulation en 6-DDL bis règle 1 : (g) Évolution de l’erreur, (h) Évolution du paramètre d’extension λ_g , (i) Évolution de la position de la caméra et (j) Évolution de l’orientation de la caméra virtuelle.

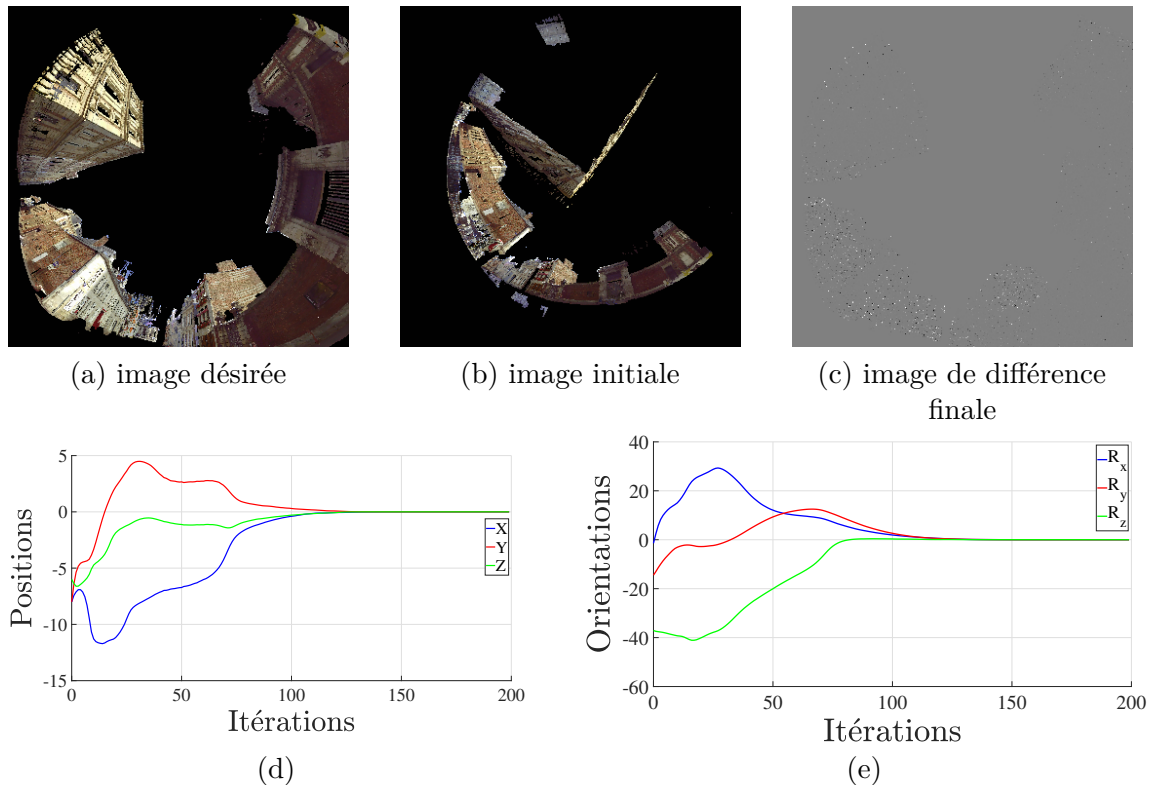


FIGURE 2.23 – Simulation en 6-DDL bis règle 2 : (d) Évolution de la position de la caméra et (e) Évolution de l'orientation de la caméra virtuelle.

2.5.7 Évaluation multiple du domaine de convergence

Nous évaluerons dans cette section le domaine de convergence des 448 poses initiales (cf. le protocole en section 2.5.1). Pour toutes les règles, le passage de l'étape 1 à l'étape 2 se produit après un nombre fixe d'itérations $n_{it} \in \mathbb{N}^*$ pour faciliter les comparaisons. Dans la suite de cette section, $n_{it} = 120$ et l'étape 2 se déroule pendant 130 itérations. Ces valeurs sont choisies pour s'assurer que chaque méthode comparée converge. De manière générale, elles ne doivent pas être considérées comme des valeurs idéales.

La figure 2.24 montre la distribution cumulative des 448 erreurs des positions finales pour les asservissements visuels photométriques, MGP selon la *Règle0*, *Règle1* et *Règle2*. En fixant un seuil de précision, nous obtenons les taux de convergence suivants :

- Pour un seuil de précision égal à 2 cm, la *Règle2* atteint un taux de réussite de 97% tandis que la *Règle1* et la *Règle0* atteignent respectivement 78% et 70%.
- Pour un seuil de convergence égal à 1 cm, la *Règle2* atteint un taux de réussite de 68% tandis que la *Règle1* et la *Règle0* atteignent respectivement 29% et 27%.

Pour la *Règle1* et la *Règle0*, l'étape 2 peut converger vers un minimum local, tandis que pour la *Règle2*, la plupart des minima locaux sont supprimés, ce qui permet d'atteindre plus

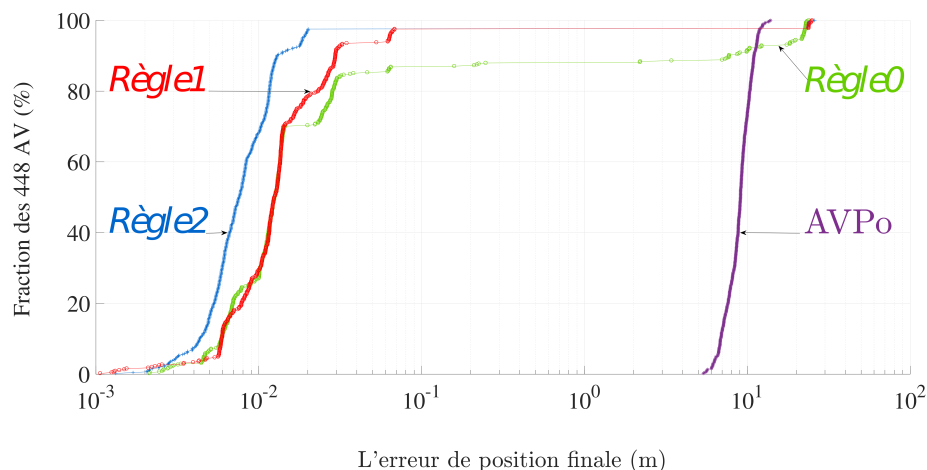


FIGURE 2.24 – Distributions cumulées des erreurs de positions finales

souvent le minimum global. En revanche, tous les AVPo (acronyme pour Asservissement visuel photométrique) divergent, ce qui n'est pas surprenant car les erreurs initiales sont supérieures à 8 m dans cette évaluation. A titre de comparaison, les auteurs dans [CARON et al., 2013] précisent que les erreurs initiales permettant aux AVPo de converger doivent être inférieures à 1,3 m. Mais, il s'agit là d'une mesure qui donne un ordre d'idée car les conditions d'expérimentation sont différentes.

Ces résultats montrent que l'asservissement visuel MGP surpasse significativement l'AVPo. De plus, la nouvelle *Règle2* surpasse aussi la *Règle0* de [CROMBEZ et al., 2019].

2.6 Conclusion

Nous avons introduit la représentation d'une image en un mélange de gaussiennes photométriques. Ce type de représentation est utilisé dans un asservissement visuel dense omnidirectionnel. Le paramètre d'extension du mélange de gaussiennes est contrôlé au même titre que la pose de caméra. Le choix des valeurs initiales de ce paramètre influe de manière importante sur le succès de l'asservissement. Afin d'améliorer ce choix, une nouvelle règle de paramétrisation a été proposée.

Plusieurs simulations ont été réalisées sur tous les degrés de liberté et avec différentes règles de paramétrisation du paramètre d'extension. Les résultats présentés montrent que l'asservissement visuel basé MGP avec la *Règle2* étend le domaine de convergence de l'asservissement visuel en assurant la meilleure précision.

Dans ce chapitre, les images courantes et désirées sont de même nature. Dans le dernier

chapitre nous généralisons l'asservissement visuel omnidirectionnel basé MGP à des images désirées capturées par une caméra réelle pour effectuer le suivi 3D par alignements successifs.

Suivi 3D par asservissements visuels virtuels successifs

Sommaire

| | | |
|------------|--|-----------|
| 3.1 | Introduction | 70 |
| 3.2 | Alignement dense direct d'images omnidirectionnelles sur nuage de points 3D | 71 |
| 3.3 | Suivi 3D par alignements successifs | 72 |
| 3.4 | Résultats expérimentaux | 73 |
| 3.4.1 | Obtention des nuages de points 3D source | 73 |
| 3.4.2 | Évaluation de la robustesse à la pose initiale | 73 |
| 3.4.3 | Application sur de longues trajectoires | 77 |
| 3.5 | Conclusion | 81 |

3.1 Introduction

L'estimation de pose relative de la caméra par rapport à la scène considérée dans cette thèse repose sur la connaissance du modèle 3D de la scène réelle. Nous pouvons donc définir le problème de calcul de pose comme étant un problème de recalage en déterminant la relation entre le modèle 3D de la scène et sa projection réelle en 2D dans le plan image. L'asservissement visuel virtuel peut être vu comme une méthode non-linéaire d'estimation de pose de caméra [SUNDARESWARAN et al., 1999, É. MARCHAND et al., 2002]. Le raisonnement suivi ici est le même que celui de l'asservissement visuel. Il consiste à faire déplacer la caméra virtuelle de sa position initiale vers la position de la caméra réelle en minimisant l'erreur entre la projection du modèle 3D dans le plan image et l'image désirée acquise par une caméra réelle. Nous supposons que la caméra virtuelle a les mêmes paramètres intrinsèques que cette dernière.

Les approches issues de l'asservissement visuel peuvent donc être adoptées pour l'asservissement visuel virtuel. En effet, la pose de la caméra réelle peut être estimée en utilisant les approches de l'asservissement visuel basées sur les primitives géométriques extraites des images synthétiques courantes issues du modèle 3D et de l'image réelle désirée. Une de ces approches a été adaptée, par exemple, à la vision omnidirectionnelle [E. MARCHAND et al., 2007]. Les approches de l'asservissement visuel basées sur une représentation complète de l'apparence de la scène réelle et virtuelle peuvent aussi être utilisées pour calculer la pose de la caméra réelle. Par exemple, dans [CARON et al., 2014], est considérée une mesure de similarité de l'information mutuelle. L'inconvénient principal de ces approches directes réside dans leur domaine de convergence étroit.

L'enjeu de ce chapitre consiste à adapter l'optimisation de la pose aux caméras omnidirectionnelles en utilisant une approche directe basée mélange de gaussiennes photométriques qui se distingue par un domaine de convergence important. Cet avantage permet d'estimer la pose de la caméra réelle avec précision, y compris lorsque l'on part d'une pose initiale éloignée de la pose désirée.

La principale difficulté réside dans la différence entre l'image de la scène observée dans les modèles virtuels et celle observée par la caméra réelle. En effet, les images courantes obtenues durant l'asservissement visuel virtuel sont le fruit de la projection d'un modèle virtuel 3D de l'environnement réel. Ce modèle est généralement obtenu par photogrammétrie et/ou lasergrammétrie. Or, l'image désirée est obtenue avec une caméra réelle dans des conditions différentes. Si l'utilisation du modèle virtuel évite le problème de la cartographie, elle engendre un problème de correspondance entre le modèle virtuel et le capteur.

Nous articulons notre raisonnement en trois temps. Nous évoquerons d'abord dans la partie 3.2, l'extension de l'asservissement visuel basé MGP à l'alignement dense direct d'image omnidirectionnelle réelle sur un nuage de points 3D. Nous traiterons ensuite du suivi 3D par alignements successifs dans la partie 3.3. Nous aborderons également dans la partie 3.4.1 les différentes méthodes pour obtenir des nuages de points 3D source. Nous rapportons enfin deux expérimentations de suivi par alignements : la première expérience avec un robot mobile

(Partie 3.4.2) et la deuxième à bord d'un véhicule (Partie 3.4.3).

3.2 Alignement dense direct d'images omnidirectionnelles sur nuage de points 3D

L'alignement dense direct d'images omnidirectionnelles sur un nuage de points 3D pourrait également être vu comme un problème d'optimisation déjà présenté dans le cas de l'asservissement visuel (Chapitre 1). Cependant, dans ce dernier cas, nous cherchions une minimisation entre deux images de synthèse issues de la même caméra virtuelle. Dans ce chapitre, l'image désirée est une image numérique réelle I_r^* . L'image virtuelle I est obtenue à partir du rendu du modèle 3D.

D'une façon similaire à une image acquise par une caméra, le rendu visuel du nuage de points 3D dépend de la position de la caméra dans l'espace ainsi que des conditions extérieures de la scène (les conditions météorologiques, l'heure, la saison, les éléments mobiles, etc.). Il est donc essentiel d'utiliser des critères qui permettent d'homogénéiser les valeurs d'intensité afin d'améliorer la correspondance entre l'image réelle I_r^* et l'image virtuelle I .

L'utilisation du critère de corrélation croisée normalisé NCC (Normalized Cross Correlation) se montre une caractéristique intéressante puisqu'il est indépendant du changement d'échelle global des intensités de l'image. Pour les applications pratiques, cela rend possible que la correspondance de deux images d'intensités différentes dues à des conditions d'éclairage variables. L'inconvénient de ce critère réside dans le fait que ses dérivées sont plus complexes à mettre en œuvre et plus longues à calculer [SUTTON et al., 2009] que celles de la SSD (Chapitre 1).

Dans le cas où les intensités entre l'image réelle et l'image virtuelle ne changent que d'un offset, ce dernier peut être résolu explicitement comme un paramètre supplémentaire dans le problème d'optimisation. Nous utilisons alors le critère de la somme des différences carrées à moyenne nulle ZSSD : Zero-mean Sum of Square Differences [SUTTON et al., 2009].

Si toutefois il y a un changement d'échelle dans la luminosité, on peut appliquer le critère de la somme normalisée de la différence au carré NSSD : Normalized Sum of Squared Differences [SUTTON et al., 2009].

Dans le cas où il y a un offset et un changement d'échelle pour la luminosité, nous pouvons considérer la somme normalisée des différences quadratiques à moyenne nulle ZNSSD : Zero-mean Normalised Sum of Squared Differences. Les intensités réelles I_r^* et les intensités virtuelles I sont centrées et normalisées avant d'être utilisées en entrée de l'algorithme d'optimisation pour améliorer leur cohérence [CARON, 2019] :

$$\forall \mathbf{u} \in \mathcal{U}, \tilde{I}_r^*(\mathbf{u}) = \frac{I_r^*(\mathbf{u}) - m_{I_r^*}}{\frac{1}{|\mathcal{U}|} \sum_{\mathbf{u}} |I_r^*(\mathbf{u}) - m_{I_r^*}|}, \quad (3.1)$$

avec

$$m_{I_r^*} = \frac{1}{\mathcal{U}} \sum_{\mathbf{u}} I_r^*(\mathbf{u}), \quad (3.2)$$

pour obtenir \tilde{I} à partir des intensités I nous suivons le même raisonnement. $\mathcal{U} = \{0, 1, \dots, P-1\} \times \{0, 1, \dots, M-1\}$, avec $P \times M$ est la taille de l'image.

Après avoir normalisé et centré l'image réelle et virtuelle, nous effectuons un alignement direct en utilisant des images omnidirectionnelles transformées en MGP et nous calculons le MGP désiré (\mathbf{G}^*) à partir de \tilde{I}^* (Fig. 3.1 - b). Le MGP courant, \mathbf{G} , est calculé à partir de \tilde{I} (Fig. 3.1 - a).

Nous notons $\tilde{\mathbf{I}}^*$, l'image réelle normalisée et centrée et $\tilde{\mathbf{I}}$, l'image virtuelle normalisée et centrée.

L'alignement dense direct d'images omnidirectionnelles sur un nuage de points 3D est basé sur la loi de commande virtuelle :

$$\mathbf{v}_g = \mu \mathbf{L}_G^+(G(\tilde{\mathbf{I}}^*) - G(\tilde{\mathbf{I}}(\mathbf{r}), \lambda_g)) \quad (3.3)$$

\mathbf{L}_G est définie en 2.10. \mathbf{v}_g est utilisé à chaque itération pour mettre à jour la pose de la caméra visuelle jusqu'à convergence.

3.3 Suivi 3D par alignements successifs

En appliquant des alignement successifs sur une séquence d'images réelles, on réalise le suivi de pose basé modèle 3D.

L'acquisition de la séquence se fait par un opérateur humain qui déplace le robot aux poses de passage. La séquence contient les images omnidirectionnelles réelles obtenues par la caméra catadioptrique embarquée sur le robot. Elles serviront d'images désirées pour les alignements successifs.

Nous appliquons le suivi de poses direct basé sur des exécutions successives de la loi de contrôle virtuel (3.3) avec la *Règle2* (voir partie 2.4.4) où chaque alignement est réalisé en deux étapes : Étape 1, $\lambda_g = \lambda_g^*$ grand avec λ_g contrôlé ; Étape 2 avec $\lambda_g = \lambda_g^* = 1$ à l'initialisation, en continuant de contrôler λ_g . La pose de la caméra virtuelle pour l'image courante acquise est initialisée avec la pose optimale de l'image précédente dans la séquence.

3.4 Résultats expérimentaux

3.4.1 Obtention des nuages de points 3D source

Afin d'obtenir un nuage de points 3D source comme modèle 3D d'un environnement réel, deux différentes technologies sont utilisées. La première consiste à utiliser un scanner laser terrestre 3D qui permet de scanner un environnement en 3 dimensions par un balayage laser (Lidar). Une fois le nuage de points obtenu, une caméra intégrée dans le scanner prend des photographies de la scène pour attribuer une couleur RVB (rouge, vert, bleu) à chaque point 3D du nuage.

La seconde technologie est la photogrammétrie. Elle consiste en une prise de photos à des emplacements variés de l'environnement afin de créer une reconstruction 3D [HÉNO et al., 2014]. Le tableau 3.1 montre les avantages et les limites des deux technologies.

3.4.2 Évaluation de la robustesse à la pose initiale

Dans cette première expérimentation, la séquence d'images omnidirectionnelles de 700×700 pixels est acquise par une caméra IDS UI-1545LE-M-GL (de cadence de 30 images par seconde) équipée d'une optique catadioptrique omnidirectionnelle *RemoteReality* (double miroir). La caméra est montée sur un Robot Mobile Pioneer 3AT piloté manuellement à la vitesse de la marche humaine (3 km/h, en moyenne) dans les rues de la ville d'Amiens. Le robot embarque un Lidar SICK LMS-200 (plan de mesure horizontal unique) utilisé par le logiciel du robot pour réaliser un SLAM qui servira d'élément de comparaison (non synchronisées) (Fig. 3.2 - a-b).

Dans cette section, on se concentre sur une partie d'environ 4,35 m d'une séquence de 350 m où le robot quitte un trottoir pour traverser une rue (Fig. 3.2 - b). Bien que courte, cette dernière partie présente une séquence de 210 images. Au début de cette dernière, un sol irrégulier conduit le robot à trembler tout en avançant en ligne droite. Même si les images sont nettes (Fig. 3.2 - c), certains mouvements inter-images sont importants en raison des secousses du robot.

Le suivi omnidirectionnel par alignement direct à partir des **M**GP (SoMGP), et à partir de l'intensité (SoP) pour comparaison, a été effectué sur cette séquence en utilisant 1 image toutes les N image(s). Différentes valeurs de N ont été utilisées afin de simuler différentes vitesses du robot (par exemple, $N = 20$ simule une vitesse de déplacement moyenne d'environ 45km/h). La figure 3.2 - a montre toutes les trajectoires estimées et celle enregistrée manuellement (SLAM Lidar), afin d'évaluer le suivi. En raison du mouvement entre les images consécutives, le SoP a du mal à suivre précisément les déplacements du robot et fournit donc des trajectoires bruitées. Le SoP est en fait fortement influencé par la valeur de N . En effet, plus N est grand, plus la dérive est importante.

| | Les avantages | Les limites |
|--|--|---|
| Numérisation à partir de la photogrammétrie | <ul style="list-style-type: none"> - Matériel avec un coût réduit et raisonnable; - Matériel accessible à tous; - Obtention du nuage de points 3D d'une manière quasi-automatique; - Mise en œuvre flexible (drone, perches...); - Possibilité d'adapter la précision (focale, distance à l'objet d'intérêt, configuration stéréoscopique). | <ul style="list-style-type: none"> - Temps de calcul important; - Ne fonctionne pas pour les surfaces non texturées; - Difficile de qualifier les résultats obtenus d'une façon systématique. |
| Numérisation par scanner laser | <ul style="list-style-type: none"> - Obtention des nuages de points 3D précis et fiables d'une façon directe; - Adapté aux grands ensembles architecturaux. | <ul style="list-style-type: none"> - Matériel avec un coût élevé; - Mise en œuvre relativement complexe qui tend à être simplifiée avec les derniers modèles plus compacts, légers; - Inaccessibilité de certaines zones; par exemple les parties extérieures hautes qui nécessitent des porteurs légers (comme un drone) qui restent rare sur le marché. - Au moment de l'acquisition, il faut que le scanner soit stable. |

TABLE 3.1 – Les avantages et les limites de la numérisation par photogrammétrie et par scanner laser [HÉNO et al., 2014].

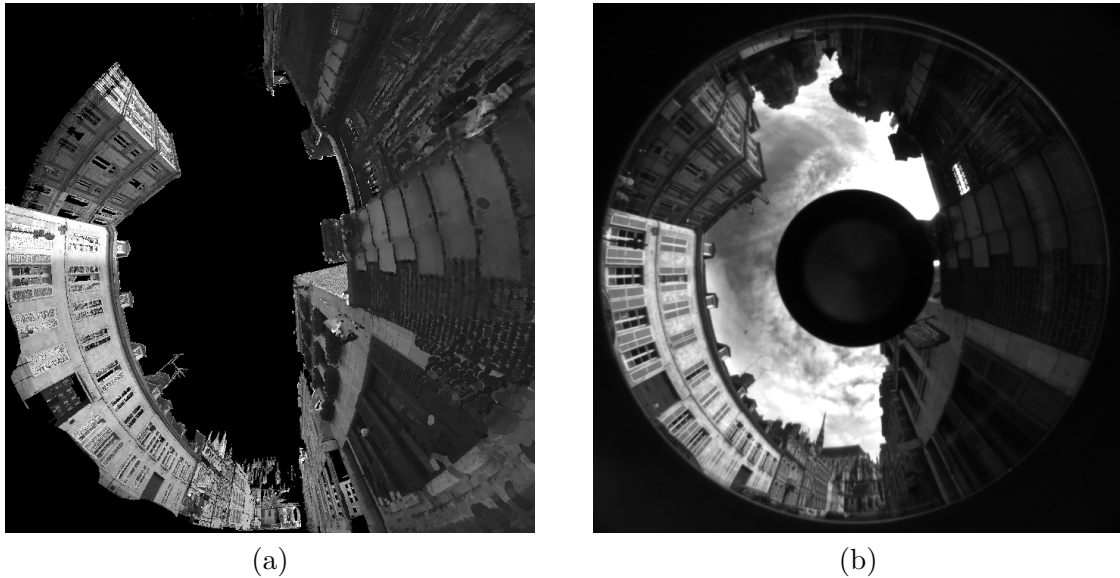
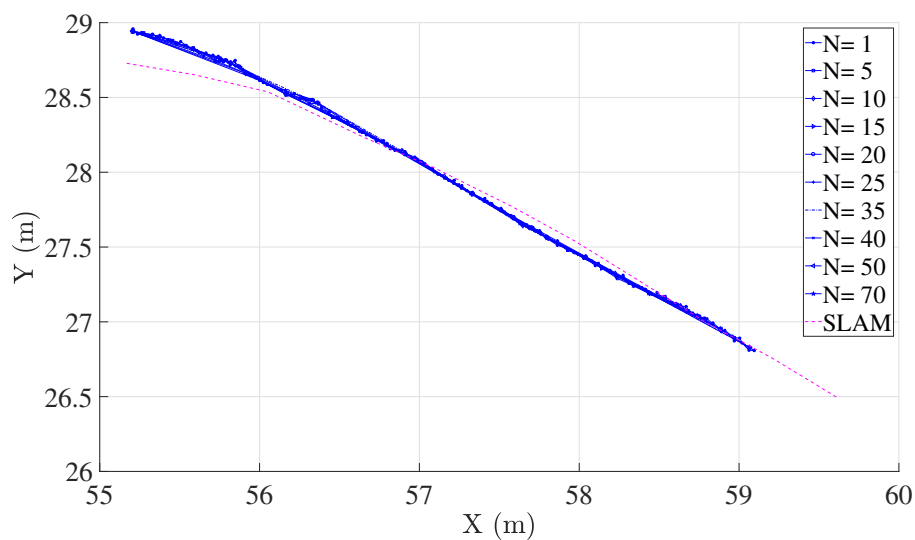


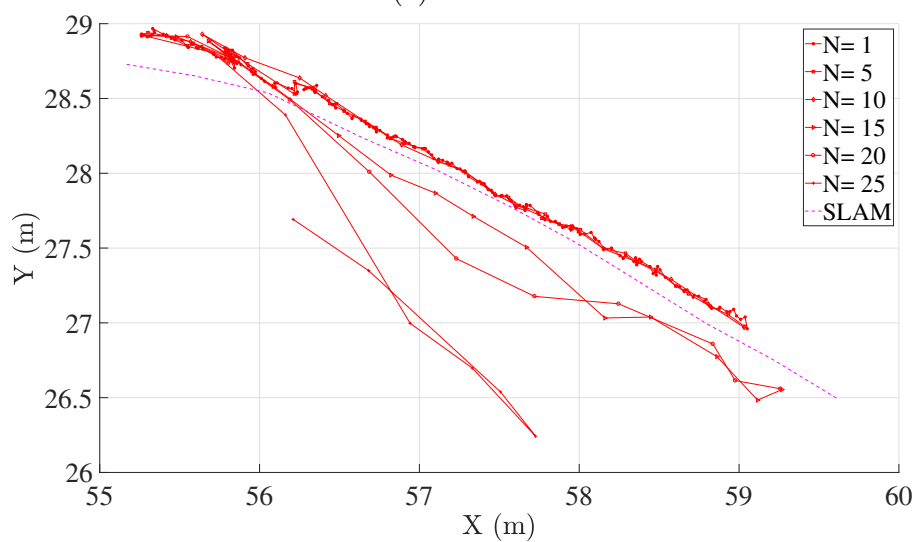
FIGURE 3.1 – (a) : Image courante issue du modèle 3D de la ville d'Amiens "*Rue Victor Hugo*", (b) : Image désirée issue d'une caméra réelle dans la même rue.

Pour $N = 25$, le SoP diverge en raison d'un mouvement inter-image trop important pour son domaine de convergence limité. En revanche, le SoMGP reste cohérent et réussit à suivre le déplacement du robot même pour des valeurs N plus élevées ($N = 70$ est équivalent à un déplacement de plus d'un mètre entre images successives). Les poses estimées avec le SoMGP sont plus fiables, plus précises et produisent des trajectoires lisses et précises qualitativement équivalentes à celles du SLAM Lidar.

La figure 3.2 - e montre la superposition des dernières images réelles et virtuelles de la séquence, rendues à la pose optimale calculée avec le SoP pour $N = 25$. Le mauvais alignement est particulièrement visible dans les zones surlignées en rouge alors que le SoMGP conduit à un alignement beaucoup plus précis (Fig. 3.2 - f).



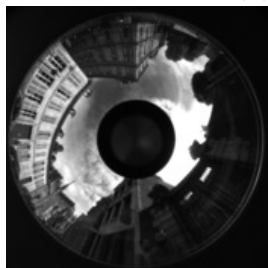
(a) SoMGP



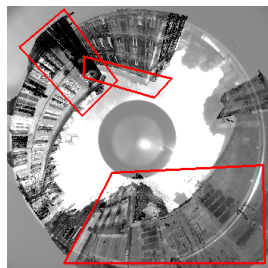
(b) SoP



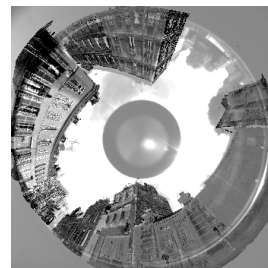
(c)



(d)



(e)



(f)

FIGURE 3.2 – Suivi direct basé sur un modèle 3D avec le SoMGP et le SoP en utilisant 1 image toutes les N images(s) : (a-b) Trajectoires estimées sur (c) un sol irrégulier et difficile où des images comme (d) sont acquises ; (e-f) Superposition des dernières images acquises et des images virtuelles optimales ($N = 25$) : (e) SoP, (f) SoMGP.

3.4.3 Application sur de longues trajectoires

Dans cette application nous utilisons un véhicule pour effectuer l'acquisition des séquences.

Le véhicule est une Peugeot 308 sur le toit de laquelle une caméra Ricoh Theta S 360 est fixée grâce à une ventouse. Pendant la conduite, les images à double-*fisheye* acquises par cette dernière caméra sont transmises à un ordinateur pour une synchronisation informatique de leur enregistrement avec les données GPS de la voiture.

Deux séquences S1 et S2 d'images brutes à double-*fisheye* sont acquises à des moments différents sur le même trajet dans les rues de la ville d'Amiens. Les images de la première séquence S1 sont transformées en images sphériques équirectangulaires utilisées en entrée du logiciel de photogrammétrie Agisoft Metashape, avec leurs coordonnées GPS comme positions initiales, pour construire hors ligne un nuage de points 3D dense. Ensuite, les images des deux séquences sont transformées en images omnidirectionnelles suivant la méthode du modèle de projection unifié¹ qui sont ensuite alignées sur le nuage de points 3D par SoMGP (et SoP pour comparaison). On peut noter que nous pourrions prendre une des deux vues *fisheye* de l'image Ricoh Theta au lieu d'en générer une autre. Cependant, l'orientation de la caméra 360 sur le toit de la voiture (imposée pour des raisons pratiques) empêche d'obtenir une vue panoramique des environs de la voiture dans une seule vue *fisheye*.

Une difficulté graduelle est obtenue en appliquant le suivi sur S1, puis sur S2. Le suivi sur S1 semble facile car les images omnidirectionnelles partagent le même point de vue que les images équirectangulaires utilisées pour construire le nuage de points 3D dense, mais le champ de vision et la non-linéarité de la résolution sont différents dans les deux types d'images, ce qui représente un premier défi. Le suivi sur S2 est plus difficile car, en plus des difficultés mentionnées pour le suivi sur S1, la trajectoire de la voiture (à la fois dans l'espace et selon la vitesse) n'est pas la même et la scène peut avoir changé entre les deux acquisitions parce que certains bus, camions ou nuages ont bougé.

La figure 3.3 montre les résultats du SoMGP (toujours avec *Règle2*) et du SoP, tous deux avec $N = 1$ car la vitesse de la voiture (22,5km/h, en moyenne) est beaucoup plus élevée que celle du robot mobile et la cadence d'acquisition des images est inférieure de 5 images par seconde. Par conséquent, le mouvement entre les images acquises successivement est parfois déjà conséquent (voir Fig. 3.4). Le SoP échoue dès que le mouvement inter-image augmente un peu (voir, par exemple, (Fig. 3.4 - a) et (Fig. 3.4 - b)) et ne suit avec succès que 6 m (1,6%) de la trajectoire de 366 m de S2 rapportée sur la figure 3.3 - b (rues "des Augustins", "Adéodat Lefèvre" et "place Saint Michel" à Amiens). Au lieu de cela, le SoMGP réussit à suivre 318 m (86,9%) de la trajectoire S2, même en présence d'un mouvement inter-image très important (voir les figures (Fig. 3.4 - c) et (Fig. 3.4 - d)). La raison de l'échec du SoMGP à la fin de la trajectoire S2 est l'absence de bâtiments sur le côté droit de la voiture qui entraîne un léger désalignement résiduel dans les images successives qui n'est jamais surmonté.

Sur S1, le SoMGP et le SoP suivent tous deux le nuage de points 3D sur des trajectoires

1. Le MIS partage son implémentation de la transformation d'images sur le dépôt <https://github.com/PerceptionRobotique/equi2omni>

plus longues (Fig. 3.3 - a) que sur S2 : SoMGP réussit à suivre sur 377 m (92,1%, soit +5,2% par rapport à S2) et le SoP sur 159 m (43,4%, soit +41,8% par rapport à S2, mais toujours bien en dessous des capacités du SoMGP). Outre que le suivi de S1 soit moins difficile en raison du fait que le nuage de points 3D est réalisé à partir d'images provenant des mêmes points de vue, S1 ne présente pas de mouvement inter-image très important, contrairement à S2 (Fig. 3.4 - d). Mais les résultats de suivi sur S1 sont présentés pour deux raisons : (i) la cohérence des poses suivies par rapport à la fois à l'empreinte des rues et à la trajectoire des poses de la caméra calculée par la photogrammétrie ; (ii) la cohérence modérée des mesures GPS en S1 (sur la figure 3.3 - a, la trajectoire GPS est parfois du mauvais côté de la rue ou sur les trottoirs ou même dans les bâtiments au niveau du virage) et son incohérence dans S2 où, au nord du virage, la trajectoire GPS est à plusieurs mètres dans les bâtiments à droite de la rue (Fig. 3.3 - b) alors que la voiture réelle ne l'est évidemment pas. Par conséquent, cette faible cohérence des mesures GPS rend inutile toute évaluation quantitative supplémentaire.

La figure 3.5 montre la superposition des dernières images capturées et des images virtuelles optimales pour le SoP, pour les deux cas : convergence (Fig. 3.5 - a) et divergence (Fig. 3.5 - b), et pour le SoMGP pour les deux cas également : convergence (Fig. 3.5 - c) et divergence (Fig. 3.5 - d).

Pour comparer, nous avons appliqué OpenVSLAM pour créer une carte à partir d'images rendues, utilisées pour localiser les images réelles. La localisation a été possible (à l'échelle) pour 12,2% de S1, soit 74,7% de moins que le SoMGP qui fournit la pose 3D à l'échelle de la caméra, donc du véhicule.



FIGURE 3.3 – Suivi direct basé sur un modèle 3D avec SoMGP et SoP avec la caméra à bord d'un véhicule. Trajectoires estimées pour (a) la première séquence S1 (GPS : bleu ; poses de la caméra issue du processus de photogrammétrie : vert foncé ; SoMGP : orange ; SoP : jaune) et (b) la deuxième séquence S2 (GPS : rouge ; SoMGP : rose ; SoP : jaune). Les trajectoires sont tracées après conversion en coordonnées de latitude et de longitude exportées sous forme de fichiers kml vers Google My Maps (carte de base monochrome).



(a) image 2732 de la S2



(b) image 2733 de la S2



(c) image 3216 de la S2



(d) image 3217 de la S2

FIGURE 3.4 – Exemples d’images successives de la séquence S2 avec (a-b) un mouvement inter-image moyen et (c-d) un mouvement inter-image important. Les images (a-b) sont acquises au début de la trajectoire (où le SoP échoue dans la figure 3.3 - b) et (c-d) dans le virage.

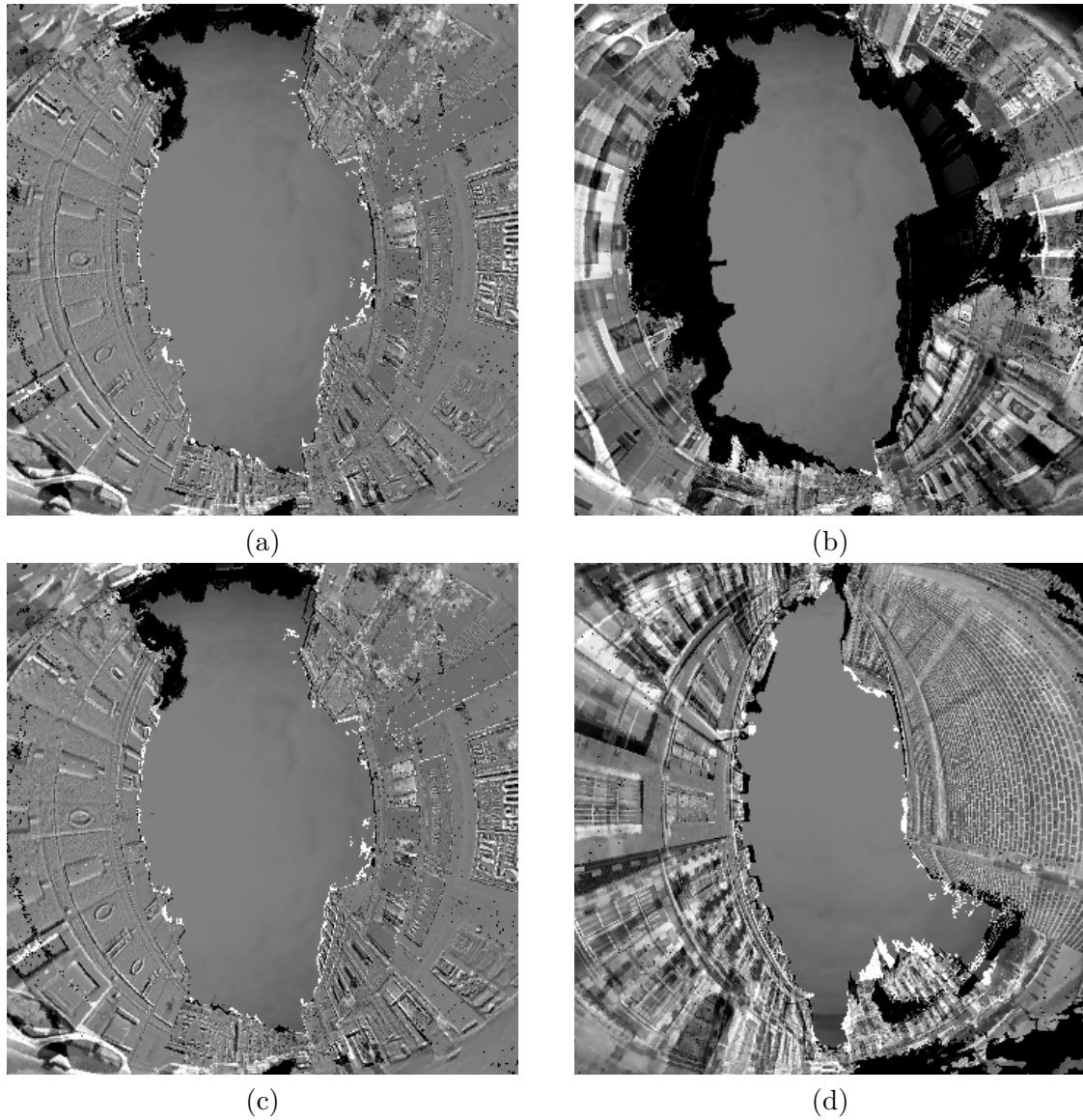


FIGURE 3.5 – Superposition des dernières images virtuelles acquises et optimales ($N = 1$) : (a) et (c) correspondent respectivement au début des trajectoires rose et jaune de la figure 3.3 - b, alors que (b) et (d) correspondent à la fin de ces trajectoires.

3.5 Conclusion

Dans les pages qui précèdent, nous avons démontré que la transformation de mélange de gaussiennes photométriques des images omnidirectionnelles comme des caractéristiques directes nous permet de résoudre le problème d'estimation de pose de caméra dans le cadre de l'asservissement visuel virtuel.

Le suivi de poses par alignements successifs a été appliqué pour deux modèles 3D de la ville d'Amiens en France. Deux expérimentations ont été réalisées avec un robot mobile et avec un véhicule. Le capteur du robot est différent de celui qui a été utilisé pour avoir le modèle 3D. Inversement, pour le véhicule deux séquences (S1 et S2) d'images issues du même capteur ont été utilisées pour construire le modèle 3D et se localiser. Le suivi omnidirectionnel virtuel par alignements successifs directs à partir des MGPs (SoMGP) et à partir de l'intensité (SoP) ont été appliqués et comparés pour différentes expérimentations et avec différentes vitesses. Le SoMGP réussit là où un important mouvement inter-image empêche le succès du SoP.

Conclusion et perspectives

La vision panoramique offre un champ de vue plus important que celui d'une caméra conventionnelle et permet ainsi de maximiser les chances d'acquérir une information visuelle riche. Rappelons d'abord que le cadre théorique retenu était celui de l'asservissement visuel virtuel et l'objectif était d'en étendre significativement le domaine de convergence, connu pour ses limites dans les approches directes. Autrement dit, il s'agissait d'optimiser un critère formulé à partir des intensités des pixels. Notre propos avait ensuite pour objet la localisation d'un robot mobile muni d'une caméra panoramique dans un modèle 3D urbain. Nous avons choisi d'aborder ce problème de localisation sous l'angle du recalage et du suivi direct basé modèle 3D dans les images panoramiques.

Dans le premier chapitre, nous avons évoqué la modélisation de la vision, et des caméras pour aller de la scène observée vers le plan pixelique. Nous avons aussi abordé le contrôle de robot par l'information visuelle captée par une caméra : l'asservissement visuel. Cette information visuelle peut être le résultat de mesures géométriques extraites des images acquises et désirées tout au long de l'asservissement visuel, ou encore utiliser toute l'information visuelle photométrique omnidirectionnelle (deuxième partie du chapitre 1). Dans le cadre de notre étude, nous nous sommes principalement intéressés à ce deuxième type d'approches.

Les limites des asservissements visuels photométriques touchent essentiellement leurs domaines de convergence. Cela nous a encouragé à combiner l'information visuelle omnidirectionnelle avec une autre représentation de l'information visuelle évoquée dans le deuxième chapitre de cette thèse : le mélange de gaussiennes photométriques. Le potentiel d'attraction, au lieu d'être concentré dans le pixel lui-même comme dans le cas photométrique, est étendu sous la forme d'une gaussienne en 2D. L'introduction du paramètre d'extension dans la loi de commande de l'asservissement visuel a permis de contrôler automatiquement l'ouverture des gaussiennes et implique un contrôle sur le taux de chevauchement entre les différentes gaussiennes de l'image.

Nous avons vu que l'asservissement visuel basé mélange de gaussiennes photométriques élargit le domaine de convergence par rapport à l'asservissement visuel purement photométrique et que l'initialisation du paramètre d'extension désiré et courant joue un grand rôle sur la convergence et la précision. En effet, un paramètre d'extension suffisamment grand permet d'élargir le domaine de convergence et un paramètre d'extension faible permet d'avoir une précision photométrique.

Nous avons proposé différentes règles pour l'initialisation du paramètre d'extension et nous

avons montré que la nouvelle *Règle2* était la meilleure. Cette règle consiste à affecter la même valeur du paramètre d'extension pour l'image initiale et désirée au début de l'asservissement pour l'étape 1, puis, pour obtenir une plus grande précision, à réduire les valeurs des deux paramètres dans l'étape 2 en faisant évoluer, au cours des deux étapes, l'extension du mélange de gaussiennes photométriques de l'image courante.

Le troisième chapitre propose un développement sur le suivi 3D par alignements successifs en utilisant des images omnidirectionnelles réelles comme points de passage. Différentes expérimentations ont été réalisées. Les suivis omnidirectionnels par alignements directs à partir des MGPs et de l'intensité ont ainsi été appliqués et comparés. Les premières expérimentations étaient réalisées avec un robot mobile et les secondes l'ont été à bord d'un véhicule. Nous avons pu voir qu'à chaque fois le suivi omnidirectionnel basé MGP réussit là où un important mouvement inter-image empêche le succès du suivi omnidirectionnel photométrique.

À l'issue de ces travaux de thèse, plusieurs perspectives de recherche sont envisageables :

- L'asservissement visuel omnidirectionnel a été réalisé dans un environnement virtuel. Même si l'alignement successif se base sur des images omnidirectionnelles réelles, un asservissement visuel réel sur un robot mobile ou un bras manipulateur est envisageable afin de valider notre approche.
- Durant nos travaux et afin d'élargir le potentiel d'attraction des différents pixels, nous nous sommes principalement concentré sur les fonctions gaussiennes. D'autres fonctions 2D avec d'autres distributions pourraient constituer une seconde perspective de recherche. Par exemple dans le cas de la diffraction de Fraunhofer, l'intensité lumineuse est calculée à partir du carré de la fonction 2D du sinus cardinal.

Financement

Ce travail est financé dans le cadre du projet INTERREG VA FMA ADAPT « Assistive Devices for empowering disAbled People through robotic Technologies ». Le Programme FMA est un programme de Coopération Territoriale Européenne qui vise à financer des projets de coopération ambitieux dans la région frontalière entre la France et l'Angleterre. Le Programme est financé par le Fonds Européen de Développement Régional (FEDER).

Bibliographie

- AHLUWALIA, R. & FOGWELL, L. (1986). A modular approach to visual servoing, In *Proceedings. 1986 IEEE International Conference on Robotics and Automation*. <https://doi.org/10.1109/ROBOT.1986.1087399>
- AL-HASSANI, S. T. (2007). *1001 inventions : Muslim heritage in our world*. Foundation for Science Technology & Civilisation, FSTC.
- ANDREFF, N., ESPIAU, B. & HORAUD, R. (2002). Visual servoing from lines. *The International Journal of Robotics Research*, 21(8), 679-699.
- BAKER, S. & NAYAR, S. K. (1999). A theory of single-viewpoint catadioptric image formation. *International journal of computer vision*, 35(2), 175-196.
- BARRETO, J. P., MARTIN, F. & HORAUD, R. (2003). Visual servoing/tracking using central catadioptric images, In *Experimental Robotics VIII*. Springer.
- BARRETO, J. P. & ARAUJO, H. (2001). Issues on the geometry of central catadioptric image formation, In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. IEEE.
- BERGOU, E., DIOUANE, Y. & KUNGURTSEV, V. (2018). Convergence and Iteration Complexity Analysis of a Levenberg–Marquardt Algorithm for Zero and Non-zero Residual Inverse Problems. Paper.
- BOLLES, R. & PAUL, R. (1973). *The use of sensory feedback in a programmable assembly system* (rapp. tech.). STANFORD UNIV CA DEPT OF COMPUTER SCIENCE.
- CARON, G. & EYNARD, D. (2011). Multiple camera types simultaneous stereo calibration, In *IEEE Int. C. on Robotics and Autom.*
- CARON, G., MARCHAND, E. & MOUADDIB, E. (2013). Photometric visual servoing for omnidirectional cameras. *Autonomous Robots*, 35(2-3), 177-193.
- CARON, G., MARCHAND, E. & MOUADDIB, E. M. (2010). Omnidirectional photometric visual servoing, In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. <https://doi.org/10.1109/IROS.2010.5651658>
- CARON, G. (2010). *Estimation de pose et asservissement de robot par vision omnidirectionnelle* (thèse de doct.). Université de Picardie Jules Verne.
- CARON, G. (2019). *Vision robotique directe* (Habilitation à diriger des recherches). Université de Picardie - Jules Verne. <https://hal.archives-ouvertes.fr/tel-02960822>
- CARON, G., DAME, A. & MARCHAND, E. (2014). Direct model based visual tracking and pose estimation using mutual information. *Image and Vision Computing*, 32(1), 54-63.
- CARON, G. & MORBIDI, F. (2018). Spherical visual gyroscope for autonomous robots using the mixture of photometric potentials, In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE.
- CHAUMETTE, F. (1990). *La relation vision-commande : théorie et application à des tâches robotiques* (thèse de doct.). Rennes 1.
- CHAUMETTE, F. & HUTCHINSON, S. (2006). Visual servo control. I. Basic approaches. *IEEE Robotics & Automation Magazine*, 13(4), 82-90.

- COLLEWET, C., MARCHAND, E. & CHAUMETTE, F. (2008a). Visual servoing set free from image processing, In *2008 IEEE International Conference on Robotics and Automation*. IEEE.
- COLLEWET, C., MARCHAND, E. & CHAUMETTE, F. (2008b). Visual servoing set free from image processing, In *2008 IEEE International Conference on Robotics and Automation*. IEEE.
- CONTICELLI, F., ALLOTTA, B. & KHOSLA, P. K. (1999). Image-based visual servoing of non-holonomic mobile robots, In *Proceedings of the 38th IEEE Conference on Decision and Control (Cat. No. 99CH36304)*. IEEE.
- COSSAIRT, O. S., MIAU, D. & NAYAR, S. K. (2011). Gigapixel Computational Imaging, In *2011 IEEE International Conference on Computational Photography (ICCP)*. <https://doi.org/10.1109/ICCPHOT.2011.5753115>
- CROMBEZ, N., MOUADDIB, E., CARON, G. & CHAUMETTE, F. (2019). Visual Servoing With Photometric Gaussian Mixtures as Dense Features. *IEEE Trans. on Robotics*, 35(1), 49-63.
- DANSEREAU, D. G., SCHUSTER, G., FORD, J. & WETZSTEIN, G. (2017). A wide-field-of-view monocentric light field camera, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- DIRAC, P. A. M. (1929). The basis of statistical quantum mechanics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 25(1), 62-66. <https://doi.org/10.1017/S0305004100018570>
- ESPIAU, B., CHAUMETTE, F. & RIVES, P. (1992). A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3), 313-326.
- FEDDEMA, J. T., LEE, C. G. & MITCHELL, O. R. (1989). Automatic selection of image features for visual servoing of a robot manipulator, In *1989 IEEE International Conference on Robotics and Automation*. IEEE Computer Society.
- FIALA, M. & BASU, A. (2002). Feature extraction and calibration for stereo reconstruction using non-svp optics in a panoramic stereo-vision sensor, In *Proceedings of the IEEE Workshop on Omnidirectional Vision 2002. Held in conjunction with ECCV'02*. IEEE.
- FLANDIN, G., CHAUMETTE, F. & MARCHAND, E. (2000). Eye-in-hand/eye-to-hand cooperation for visual servoing, In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. IEEE.
- FOMENA, R. T. (2008). *Asservissement visuel par projection sphérique* (thèse de doct.). Université Rennes 1.
- GEYER, C. & DANIILIDIS, K. (2000). A unifying theory for central panoramic systems and practical implications, In *European conference on computer vision*. Springer.
- GONZÁLEZ, R. C. & WOODS, R. E. (1992). *Digital image processing*. Addison-Wesley.
- GUERBAS, S. E., CROMBEZ, N., CARON, G. & MOUADDIB, E. M. (2021). Mélange de Gaussiennes Photométriques pour l'Asservissement Visuel Virtuel Direct d'une Caméra Omnidirectionnelle, In *ORASIS 2021*, Saint Ferréol, France. Centre National de la Recherche Scientifique [CNRS]. <https://hal.archives-ouvertes.fr/hal-03339652>
- GUERBAS, S. E., CROMBEZ, N., CARON, G. & MOUADDIB, E. M. (2021). Direct 3D model-based tracking in omnidirectional images robust to large inter-frame motion, In *2021*

- 20th International Conference on Advanced Robotics (ICAR)*. <https://doi.org/10.1109/ICAR53236.2021.9659324>
- GUERBAS, S.-E., CROMBEZ, N., CARON, G. & MUSTAPHA MOUADDIB, E. (2021). Photometric Gaussian Mixtures for Direct Virtual Visual Servoing of Omnidirectional Camera, In *IEEE CVPR 2021 Workshop on 3D Vision and Robotics*, Seattle (virtual), United States. <https://hal.archives-ouvertes.fr/hal-03227374>
- HADJ-ABDELKADER, H. (2006). *Asservissement visuel en vision omnidirectionnelle* (thèse de doct.). Université Blaise Pascal-Clermont-Ferrand II.
- HAÏDAR, R. (2021). Abu Ali Al-Hasan Ibn Al-Hasan Ibn Al-Haytham, dit Alhazen. In *Sous la lumière, les hommes* (p. 28-33). EDP Sciences. <https://doi.org/doi:10.1051/978-2-7598-1205-9-007>
- HÉNO, R. & CHANDELIER, L. (2014). *Numérisation 3D de bâtiments : Cas des édifices remarquables*. ISTE editions. <https://books.google.fr/books?id=bnCnoAEACAAJ>
- HONG, J., TAN, X., PINETTE, B., WEISS, R. & RISEMAN, E. M. (1992). Image-based homing. *IEEE Control Systems Magazine*, 12(1), 38-45.
- HORAUD, R., DORNAIKA, F. & ESPIAU, B. (1998). Visually guided object grasping. *IEEE Transactions on Robotics and Automation*, 14(4), 525-532.
- HORN, B. K. & SCHUNCK, B. G. (1981). Determining optical flow, In *Techniques and Applications of Image Understanding*. International Society for Optics et Photonics.
- HUGHES, C., DENNY, P., JONES, E. & GLAVIN, M. (2010). Accuracy of fish-eye lens models. *Applied optics*, 49(17), 3338-3347.
- JARVIS, R. (1977). A Simple Hand/Eye Experimental Set-up for Computer Vision Research. *Computer*, 10(9), 65-69. <https://doi.org/10.1109/C-M.1977.217864>
- LAI, J. K. L. & CHENG, B. A. (2011). Capturing Two or More Images to Form a Panoramic Image [US Patent App. 12/730,628]. Google Patents.
- MA, Y., SOATTO, S., KOŠECKÁ, J. & SASTRY, S. (2004). *An invitation to 3-d vision : from images to geometric models* (T. 26). Springer.
- MARCHAND, E. & CHAUMETTE, F. (2007). Fitting 3D Models on Central Catadioptric Images, In *Proceedings 2007 IEEE International Conference on Robotics and Automation*. <https://doi.org/10.1109/ROBOT.2007.363764>
- MARCHAND, E. & CHAUMETTE, F. (2005). Feature tracking for visual servoing purposes. *Robotics and Autonomous Systems*, 52(1), 53-70.
- MARCHAND, É. & CHAUMETTE, F. (2002). Virtual Visual Servoing : a framework for real-time augmented reality, In *Computer Graphics Forum*. Wiley Online Library.
- MARCHAND, É., SPINDLER, F. & CHAUMETTE, F. (2005). ViSP for visual servoing : a generic software platform with a wide class of robot control skills. *IEEE Robotics & Automation Magazine*, 12(4), 40-52.
- MARHIC, B., MOUADDIB, E. & PEGARD, C. (1998). A localisation method with an omnidirectional vision sensor using projective invariant, In *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*. <https://doi.org/10.1109/IROS.1998.727442>

- MEZOUAR, Y. & MALIS, E. (2004). Robustness of central catadioptric image-based visual servoing to uncertainties on 3D parameters, In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566)*. IEEE.
- MIYAMOTO, K. (1964). Fish eye lens. *JOSA*, 54(8), 1060-1061.
- NAYAR, S. K. (1997a). Catadioptric omnidirectional camera, In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*. IEEE.
- NAYAR, S. K. (1997b). Omnidirectional video camera, In *Proc. DARPA Image Understanding Workshop*. Citeseer.
- PEGARD, C. & MOUADDIB, E. (1996). A mobile robot using a panoramic view, In *Proceedings of IEEE International Conference on Robotics and Automation*. <https://doi.org/10.1109/ROBOT.1996.503578>
- PEGARD, C. & MOUADDIB, E. M. (1996). A mobile robot using a panoramic view, In *Proceedings of IEEE International Conference on Robotics and Automation*. IEEE.
- PIERCE, D., HERRNSTADT, S. M., GROVER, T. & GOOD, F. (2005). Stereoscopic panoramic image capture device [US Patent 6,947,059]. Google Patents.
- RASMUSSEN, C. E. Et al. (1999). The infinite Gaussian mixture model., In *NIPS*.
- REES, D. W. (1970). Panoramic television viewing system [US Patent 3,505,465]. Google Patents.
- RUF, A. & HORAUD, R. (2000). Vision-based guidance and control of robots in projective space, In *European Conference on Computer Vision*. Springer.
- SHANNON, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3), 379-423.
- SILVEIRA, G., MIRISOLA, L. & MORIN, P. (2020). Decoupled Intensity-Based Nonmetric Visual Servo Control. *IEEE Transactions on Control Systems Technology*, 28(2), 566-573. <https://doi.org/10.1109/TCST.2018.2873541>
- SMITH, A. M. (2001). *Alhacen's Theory of Visual Perception : A Critical Edition, with English Translation and Commentary, of the First Three Books of Alhacen's De Aspectibus, the Medieval Latin Version of Ibn Al-Haytham's Kitab Al-Manazir* (T. 1). American Philosophical Society.
- SMITH, J. D. (1992). The remarkable Ibn al-Haytham. *The Mathematical Gazette*, 76(475), 189-198. <https://doi.org/10.2307/3620392>
- SOUTHWELL, D., BASU, A., FIALA, M. & REYDA, J. (1996). Panoramic stereo, In *Proceedings of 13th International Conference on Pattern Recognition*. IEEE.
- STURM, P. (2002). Mixing catadioptric and perspective cameras, In *Proceedings of the IEEE Workshop on Omnidirectional Vision 2002. Held in conjunction with ECCV'02*. IEEE.
- SUNDARESWARAN, V. & BEHRINGER, R. (1999). Visual servoing-based augmented reality, In *IWAR'98 : Proceedings of the international workshop on Augmented reality : placing artificial objects in real scenes*, AK Peters, Ltd., Natick, MA, USA.
- SUTTON, M. A., ORTEU, J. J. & SCHREIER, H. (2009). *Image correlation for shape, motion and deformation measurements : basic concepts, theory and applications*. Springer Science & Business Media.
- XIONG, Y. & TURKOWSKI, K. (1997). Creating image-based VR using a self-calibrating fisheye lens, In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*. IEEE.

-
- YAGI, Y. & KAWATO, S. (1990). Panorama scene analysis with conic projection, In *EEE International Workshop on Intelligent Robots and Systems, Towards a New Frontier of Applications*. IEEE.
- YAGI, Y., KAWATO, S. & TSUJI, S. (1994). Real-time omnidirectional image sensor (COPIS) for vision-guided navigation. *IEEE Transactions on robotics and automation*, 10(1), 11-22.

Résumé — La vision omnidirectionnelle capture dans toutes les directions une scène en temps réel grâce à un champ de vision plus étendu que celui offert par une caméra conventionnelle. Au sein de l’environnement, relier les caractéristiques visuelles contenues dans les images de la caméra à ses mouvements est une problématique centrale pour l’asservissement visuel. Les approches directes se caractérisent cependant par un domaine de convergence limité. La thèse que nous présentons a pour premier objectif d’étendre significativement ce dernier dans le cadre de l’asservissement visuel virtuel en représentant l’image omnidirectionnelle par un Mélange de Gaussiennes Photométriques (MGP). Cette approche est étendue dans un deuxième temps au recalage et au suivi direct basé modèle 3D dans les images omnidirectionnelles. Cela permet d’étudier la localisation d’un robot mobile équipé d’une caméra panoramique dans un modèle urbain 3D. Les expérimentations ont été conduites en environnement virtuel et en utilisant des images réelles capturées à l’aide d’un robot mobile et d’un véhicule. Les résultats montrent un agrandissement significatif du domaine de convergence qui permet alors une grande robustesse face à d’importants mouvements inter-images.

Mots clés : Asservissement visuel virtuel, mélange de gaussiennes photométriques, suivi visuel, estimation de pose, robotique.

Abstract —

Omnidirectional vision captures a scene in real-time in all directions with a wider field of view than a conventional camera. Within the environment, linking the visual features contained in the camera images to its movements is a central issue for visual servoing. Direct approaches, however, are characterized by a limited range of convergence. The main objective of this dissertation is to significantly extend the area of convergence in the context of virtual visual servoing by representing the omnidirectional image by a Photometric Gaussian Mixtures (PGM). This approach is further extended in the second step to the registration and direct tracking based on 3D models in omnidirectional images. This proposed methodology allows for studying the localization of a mobile robot equipped with a panoramic camera in a 3D urban model. The results show a significant enlargement of the convergence domain for high robustness to large interframe movements, as evidenced by experiments in virtual environments and with real images captured with a mobile robot and a vehicle.

Keywords : Virtual visual servoing, photometric gaussian mixtures, visual tracking, pose estimation, robotics.
