UNIVERSITÉ de Picardie Jules Verne

Thèse de Doctorat

Mention …….Sciences pour l'Ingénieur
Spécialité …….Vision par Ordinateur

présentée à

l'École Doctorale en Sciences Technologie et Santé (ED 585)

de l'Université de Picardie Jules Verne

par

Nathan Crombez

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

Contributions aux asservissements visuels denses : nouvelle modélisation des images adaptée aux environnements virtuels et réels

Soutenue le 09 décembre 2015 après avis des rapporteurs, devant le jury d'examen :

M. François CHAUMETTE, Directeur de recherche	**Rapporteur**
M. Youcef MEZOUAR, Professeur	**Rapporteur**
M. Raja CHATILA, Professeur	**Examinateur**
M. Philippe MARTINET, Professeur	**Examinateur**
M. Claude PÉGARD, Professeur	**Examinateur**
M. Guillaume CARON, Maître de conférence	**Examinateur**
M. El Mustapha MOUADDIB, Professeur	**Examinateur**

Remerciements

Comme il est d'usage, je vais entreprendre la délicate étape qu'est la rédaction des remerciements. La difficulté ne réside pas dans le fait de faire connaître ma gratitude envers tout ceux qui m'ont apporté leur soutien et leur aide tout au long de ces trois dernières années mais plutôt de m'assurer de n'omettre personne. C'est pourquoi, je tiens à remercier par avance tout ceux que je vais oublier et dont le nom ne figurera par sur cette page. Cet exercice est doublement délicat car son écriture marque également la fin de la rédaction de ce manuscrit. C'est donc avec une certaine émotion et beaucoup de sincérité que j'écris les dernières lignes de cette thèse.

Dans un premier temps, je tiens à remercier très chaleureusement mes encadrants Guillaume Caron et El Mustapha Mouaddib pour l'aide précieuse qu'ils n'ont eu de cesse de m'apporter. J'ai énormément appris à leurs côtés et leur en suis infiniment reconnaissant. L'encadrement scientifique, la bonne humeur et la disponibilité qu'ils ont eu à mon égard m'ont permis de m'épanouir pleinement dans mes travaux. Je les remercie également très sincèrement de m'avoir conseillé et rassuré lorsque j'en avais besoin, tant d'un point de vue scientifique que sur le plan humain.

Je tiens à remercier vivement Claude Pégard, mon directeur de thèse. Même si nous n'avons pas pu travailler ensemble, ses conseils avisés, sa disponibilité et sa bonne humeur permanente ont été une grande source de motivation pour mener à bien mes recherches.

Je souhaite également exprimer ma profonde gratitude envers mon jury de thèse, à commencer par Raja Chatila, qui m'a fait l'honneur de le présider. Je remercie mes rapporteurs, François Chaumette et Youcef Mezouar, d'avoir accepté de lire et d'évaluer le manuscrit. Grâce à leur grande expertise dans la communauté vision et robotique, les critiques et commentaires de leur rapport très détaillé et constructif m'ont permis d'améliorer significativement la qualité du manuscrit. Mes remerciements vont également à Raja Chatila et Philippe Martinet pour avoir examiné avec rigueur mes travaux de thèse. Je les remercie de s'être déplacés pour ma soutenance et pour l'intérêt qu'ils ont démontré au travers de leurs nombreuses questions.

J'aimerais également remercier l'ensemble des membres du MIS qui, grâce à leur bonne humeur et leur gentillesse, font du laboratoire un endroit où il fait bon travailler. Je pense notamment aux petits déj' du vendredi, à la chandeleur ou encore aux repas de noël pendant lesquels la convivialité était de mise.

Bien entendu, sans tous les citer, j'ai une pensée pour tous les doctorants, postdocs et stagiaires que j'ai pu rencontrer au cours de ces trois années. Merci tout d'abord aux doctorants qui commençaient leur dernière année lors de mon arrivée au laboratoire. Je pense notamment à Fatima Zahra Benamar, Ibrahim

Abdi et Romain Marie qui, même avec ses écouteurs sur les oreilles, nous faisait profiter de ses goûts musicaux plus que discutables. Je tiens à remercier tout particulièrement Zaynab Habibi. Sa thèse ayant commencé en même tant que la mienne, nous nous sommes entraidés et encouragés mutuellement tout au long de ces trois années ("Wili Wili Wili !"). Je pense également à Nicolas Guiomard-Kagan sans qui les pauses cafés auraient été très (trop) calmes et les repas moins long ! Comment parler de Nicolas sans mentionner Youssef Alj et Clément Lecat qui, eux aussi, n'étaient pas avares de paroles lorsqu'il s'agissait de débattre sur des sujets divers et variés. Je songe enfin à Cyril Séguin, Noureddine Mohtaram, Mohamed Ouddaf, Doha El Hellani, Hanane Barramou et tout ceux que j'oublie mais qui se reconnaîtront.

Je remercie grandement les personnes de mon entourage, famille et amis, pour leur soutien, leur aide (tout particulièrement : merci soeurette !) et pour m'avoir régulièrement rappeler qu'il existe une vie en dehors de la thèse. Enfin, merci infiniment à Isabelle de m'avoir supporté (dans les deux sens du terme) tout au long de ces trois années.

Table des matières

Table des figures

Liste des tableaux

Préambule

De la robotique à la réalité augmentée en passant par la réalité virtuelle, les caméras numériques sont de plus en plus utilisées compte-tenu de la richesse des informations perçues par ce type de capteur.

Cette thèse aborde l'estimation de pose et le positionnement de caméra sous le formalisme de l'asservissement visuel. L'asservissement visuel est une méthode de contrôle en boucle fermée des mouvements d'un système dynamique en utilisant des données visuelles comme retour d'informations. Les données visuelles peuvent être acquises à l'aide d'une ou de plusieurs caméras numériques. Cette ou ces caméras peuvent être mises en mouvement par le système ou elles peuvent être fixées dans l'espace de travail afin d'observer l'environnement à partir de points de vue extérieurs au système. Dans nos travaux, nous nous intéresserons tout particulièrement au cas monoculaire où la caméra est directement déplacée par le système dans l'environnement.

Une représentation virtuelle de l'environnement peut être créée à partir de points ou de droites 3D caractéristiques ou sous la forme d'un modèle 3D complet. Il est alors possible de retrouver la pose d'une caméra numérique à l'origine d'une image réelle à partir de caractéristiques visuelles provenant d'images virtuelles générées dans ces environnements. C'est ce que l'on appelle l'asservissement visuel virtuel.

Que la caméra soit réelle ou virtuelle, il est essentiel d'établir une relation entre les caractéristiques visuelles provenant de ses images et ses propres déplacements dans l'environnement. Généralement, les caractéristiques visuelles utilisées sont géométriques (points, lignes, etc.). Cependant, des étapes de traitements des images sont nécessaires pour extraire, suivre ou encore segmenter ce type de caractéristiques. Bien qu'ayant été, et étant toujours, très étudiées, ces étapes restent très délicates. À tel point que les résultats des asservissements visuels basés sur ce type de caractéristiques sont étroitement liés à la qualité d'extraction de ces mesures dans les images. C'est pourquoi, il a été proposé d'exploiter directement l'apparence de la scène plutôt que des mesures éparses censées la représenter. L'utilisation de la totalité des intensités des pixels des images comme caractéristique visuelle dense apporte une redondance d'informations et permet de passer outre aux problèmes liés à la détection, la mise en correspondance, le suivi ou encore la segmentation de l'image.

Sans compter ce préambule et la conclusion générale, le manuscrit de thèse se structure en quatre chapitres. Le premier chapitre énonce tout d'abord quelques notations et notions fondamentales sur la vision par ordinateur. Les différents modèles de caméra utilisés dans cette thèse ainsi que les relations géométriques

entre la caméra et l'environnement dans lequel elle évolue sont présentés. S'ensuit un état-de-l'art non-exhaustif répertoriant les différentes caractéristiques visuelles employées en asservissement visuel. Ces caractéristiques sont globalement classées en deux groupes : les géométriques et les photométriques. Enfin, les différents types de modèle 3D exploités en asservissement visuel virtuel sont présentés.

Le deuxième chapitre démontre comment la caractéristique visuelle purement photométrique peut être étendue à l'asservissement visuel virtuel. Bien entendu, pour être en mesure d'utiliser cette primitive visuelle, le modèle 3D virtuel doit être doté d'informations photométriques représentant l'environnement. Ce qui nous amène à présenter le programme de recherche E-Cathédrale qui est à l'origine du modèle 3D utilisé dans nos travaux. Ce modèle est composé de nuages de points colorés acquis à l'aide de scanners laser 3D. L'estimation de pose de caméra à l'aide de ce modèle sous le formalisme de l'asservissement visuel virtuel est développée puis est mise en situation dans deux applications concrètes, mettant en oeuvre caméras perspective et omnidirectionnelle.

Les différentes expérimentations menées dans le deuxième chapitre ont permis de mettre en lumière les limites des asservissements visuels purement photométriques. En effet, la pose optimale de la caméra étant considérée comme la solution d'un problème d'optimisation non-linéaire, la convergence vers cette solution dépend directement de la distance entre la pose initiale de la caméra et la pose désirée. C'est pourquoi, le chapitre 3 propose une nouvelle modélisation des images qui permet d'agrandir le domaine de convergence des asservissements visuels. Il s'agit de représenter l'image sous la forme d'un mélange de gaussiennes photométriques. Cette nouvelle modélisation est tout d'abord exploitée en asservissement visuel, c'est-à-dire lorsque l'image désirée et les images acquises durant l'optimisation sont issues d'une même caméra. Puis, dans le dernier chapitre, nous revenons à l'asservissement visuel virtuel mais en utilisant cette nouvelle modélisation des images.

Le quatrième et dernier chapitre réunit les contributions issues des deux chapitres précédents. La nouvelle modélisation des images est étendue à l'asservissement visuel virtuel. Différentes formulations de cette modélisation sont proposées et comparées. Les études menées et les résultats obtenus dans ce chapitre indiquent clairement que la nouvelle modélisation des images se prête tout particulièrement bien aux modèles 3D provenant de scanners laser, comblant ainsi un manque de l'état de l'art.

Enfin, la conclusion met en avant les contributions de la thèse et énonce aussi quelques perspectives de recherche.

Chapitre 1

Introduction

Sommaire

1.1 Notations et notions de base

1.1.1 Modélisation de caméras

L'évolution de la vision par ordinateur a suivi un chemin semblable à de nombreux domaines scientifiques. Le postulat de base étant que l'être humain sait parfaitement percevoir et interagir avec le monde 3D qui l'entoure, les premières caméras ont un système de construction de l'image algorithmiquement semblable à celui de l'œil humain. Par la suite, des systèmes de vision plus complexes ont vu le jour afin de percevoir différemment le monde 3D environnant.

1.1.1.1 Caméra Perspective

Le modèle de projection se rapprochant le plus de celui de l'œil humain est le modèle perspectif. La projection perspective est le type de projection le plus communément utilisé en vision par ordinateur. La géométrie de transformation du monde 3D vers l'image 2D d'une caméra perspective est similaire à la formation d'une image dans une chambre noire (camera obscura en latin). Une chambre noire est constituée d'une pièce plongée dans l'obscurité dont l'une des

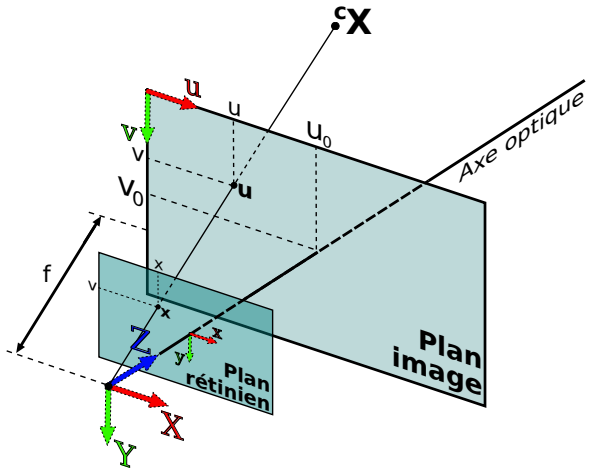


FIGURE 1.1: Schéma de la projection perspective d'un point du monde 3D dans l'image 2D

parois est percée d'un minuscule trou. Les rayons lumineux réfléchis dans toutes les directions par les objets du monde et entrant dans la chambre sont contraints à entrer par l'unique trou de celle-ci. Par conséquent, chaque point de la surface du mur à l'intérieur de la chambre faisant face au trou ne reçoit la lumière que d'un seul rayon (dans le cas idéal) projeté par un point précis d'un objet. Il se forme donc sur le mur une image inversée de la scène extérieure faisant face au trou de la chambre.

Une caméra perspective suit les mêmes principes que la chambre noire, le trou de convergence des rayons lumineux est appelé centre de projection et le mur où se forme l'image est un capteur photosensible.

La transformation d'un point 3D ${}^{\mathbf{c}}\mathbf{X} = ({}^{c}X, {}^{c}Y, {}^{c}Z, 1)^T$ dans le repère caméra ${}^{c}\mathcal{R}$ vers un point 2D $\mathbf{x} = (x, y, 1)^T$ dans le repère capteur photosensible ${}^{r}\mathcal{R}$ suit la projection perspective $pr_p()$ suivante :

$$\mathbf{x} = pr_p({}^{\mathbf{c}}\mathbf{X}) \quad avec \quad \begin{cases} x = f\dfrac{{}^{c}X}{{}^{c}Z} \\ y = f\dfrac{{}^{c}Y}{{}^{c}Z} \end{cases} . \tag{1.1}$$

f correspond à la distance focale séparant le centre de projection et le capteur photosensible. Le point $\mathbf{x}$ appartient au plan rétinien exprimé dans le repère capteur ${}^{r}\mathcal{R}$ en unité métrique.

Une transformation supplémentaire est nécessaire pour obtenir une image numérique d'une scène telle que nous la connaissons ordinairement. En effet, une image digitale est le fruit de la discrétisation spatiale de l'image formée sur le plan rétinien par un échantillonnage régulier. Cet échantillonnage est composé de $M \times N$ pixels (picture elements) avec M la largeur de l'image digitale et N

sa hauteur. La transformation permettant de passer du repère capteur ${}^{r}\mathcal{R}$ au repère image ${}^{i}\mathcal{R}$ pour un point $\mathbf{x}$ est :

$$\mathbf{u} = \mathbf{K}\mathbf{x} \tag{1.2}$$

où $\mathbf{u} = (u, v)$ est la position du point $\mathbf{x}$ dans le plan image en coordonnées pixeliques. La matrice de passage $\mathbf{K}$ est la matrice des paramètres intrinsèques de la caméra :

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

dans laquelle u_0 et v_0 désignent les coordonnées du point principal dans l'image. Ce point principal correspond à l'intersection de l'axe optique avec le plan image. α_u et α_v contiennent k_x et k_y le nombre de pixels par unité de longueur suivant respectivement les directions x et y du capteur, tout en prenant en compte la focale f de la caméra.

Finalement, le modèle de projection perspectif complet permettant de projeter un point 3D exprimé dans le repère caméra ${}^{c}\mathcal{R}$ au plan image pixelique dans le repère ${}^{i}\mathcal{R}$ peut s'écrire :

$$\mathbf{u} = \mathbf{K} pr_p({}^{\mathbf{c}}\mathbf{X}) \tag{1.4}$$

La Figure 1.2 montre un exemple de deux images numériques perspectives prises à l'intérieur de la cathédrale d'Amiens. À titre de comparaison, dans ce qui suit, des images numériques ont été prises à partir de cette même position mais en utilisant d'autres types de caméra (Figure 1.3 et Figure 1.7).

FIGURE 1.2: Images numériques perspectives de l'intérieur de la cathédrale d'Amiens (E-Cathédrale)

Il est possible de retrouver les coordonnées 3D d'un point à partir de ses coordonnées image à condition de connaître la profondeur, c'est-à-dire la distance entre ce point 3D et le centre optique de la caméra.

Une caméra est dite calibrée lorsque ses paramètres intrinsèques sont connus. La symétrie sphérique des lentilles des caméras peut être à l'origine de distorsions radiales et tangentielles. La première est due à l'asymétrie des lentilles, la seconde à un mauvais alignement des lentilles. Les paramètres de distorsion peuvent également être estimés et utilisés afin d'enrichir le modèle de projection.

Les caméras perspectives ont été et sont toujours très utilisées en vision par ordinateur ou en robotique. Cependant la limitation de leur champ de vue engendre un manque d'information perceptible. C'est pourquoi différentes solutions ont été imaginées dans le but d'augmenter considérablement le champ de vision pour le rendre panoramique, voire sphérique.

1.1.1.2 Caméra Omnidirectionnelle

Le champ de vue des caméras perspectives étant limité, plusieurs méthodes permettant d'accroître le champ de vision perceptible par une caméra ont été proposées. Nous nous intéressons dans cette section aux méthodes utilisant une seule caméra et capable d'obtenir une image omnidirectionnelle avec une seule prise de vue.

Une première approche consiste à utiliser un objectif très grand angle plus communément appelé fisheye. Les objectifs de type fisheye possèdent une lentille spéciale très incurvée. La distance focale très courte permet d'obtenir un angle de champ très grand, pouvant atteindre 180° avec une perception à 360° autour de l'axe de la caméra. La Figure 1.3 montre un exemple d'image numérique obtenue avec un objectif de type fisheye à l'intérieur de la cathédrale d'Amiens. Une seconde approche consiste à associer une caméra conventionnelle à un miroir de révolution. Ce type de dispositif est appelé catadioptrique. Le monde entourant le dispositif est réfléchi par le miroir vers l'optique de la caméra placée face à celui-ci. La contrainte du point de vue unique implique que les rayons réfléchis par le miroir convergent vers un même point.

Un modèle de projection a été proposé afin d'unifier les caméras à point de vue unique [Geyer 2000, Barreto 2001]. Même si la caméra fisheye n'est pas à point de vue unique, ce modèle unifié avec prise en compte des paramètres de distorsion est une bonne approximation de la projection de ce type de caméra [Ying 2004] [Courbon 2007].

D'après le modèle de projection unifié [Barreto 2001], la génération de l'image d'une scène peut s'écrire comme étant la combinaison d'une projection sphérique et d'une projection sur un plan. Dans un premier temps, la projection sphérique $pr_s()$ projette un point 3D ${}^{\mathbf{c}}\mathbf{X} = ({}^{c}X, {}^{c}Y, {}^{c}Z, 1)^T$ appartenant au repère caméra

FIGURE 1.3: Image numérique fisheye de l'intérieur de la cathédrale d'Amiens (E-Cathédrale)

${}^{c}\mathcal{R}$ sur une sphère unitaire en ${}^{\mathbf{s}}\mathbf{X} = ({}^{s}X, {}^{s}Y, {}^{s}Z, 1)^{T}$.

$$ {}^{s}\mathbf{X} = pr_{s}({}^{\mathbf{c}}\mathbf{X}) \quad avec \quad \begin{cases} {}^{s}X = \dfrac{{}^{c}X}{\rho} \\ {}^{s}Y = \dfrac{{}^{c}Y}{\rho} \\ {}^{s}Z = \dfrac{{}^{c}Z}{\rho} \end{cases} . \tag{1.5} $$

où $\rho = \sqrt{{}^{c}X^{2} + {}^{c}Y^{2} + {}^{c}Z^{2}}$ correspond à la distance entre le point 3D et le centre de la sphère. La seconde étape consiste à projeter les points de la sphère unitaire sur le plan rétinien par une projection perspective : $\mathbf{x} = pr_{p}({}^{s}\mathbf{X})$. Dans le repère caméra, la sphère unitaire est centrée en $(0, 0, \xi)^{T}$, ξ correspond donc à la distance entre le centre de la sphère et le centre optique de la caméra. Cette distance est généralement appelée l'excentricité du miroir. La combinaison de la projection sphérique et de la projection perspective permet de lier directement $\mathbf{x}$ et ${}^{s}\mathbf{X}$ et

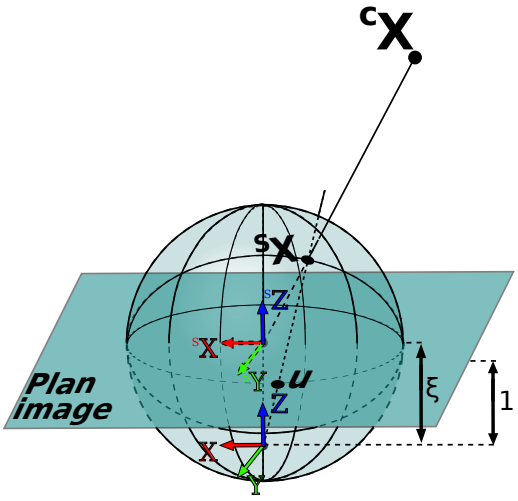


FIGURE 1.4: Schéma de la projection unifiée d'un point du monde 3D dans l'image 2D en passant par la sphère unitaire [Barreto 2001]

nous donne le modèle de projection générique $pr_o()$ suivant :

$$\mathbf{x} = pr_o({}^{\mathbf{c}}\mathbf{X}) \quad avec \quad \begin{cases} x = \dfrac{{}^cX}{{}^cZ + \xi\rho} \\ y = \dfrac{{}^cY}{{}^cZ + \xi\rho} \end{cases} . \tag{1.6}$$

Comme précédemment le point $\mathbf{x}$ appartient au plan rétinien, ses coordonnées pixeliques $\mathbf{u}$ sont obtenues à l'aide de la matrice des paramètres intrinsèques $\mathbf{K}$ de la caméra (eq. 1.3). Par conséquent, le modèle de projection omnidirectionnelle complet permettant de projeter un point 3D exprimé dans le repère caméra ${}^c\mathcal{R}$ au plan image pixelique dans le repère ${}^i\mathcal{R}$ peut s'écrire :

$$\mathbf{u} = \mathbf{K} pr_o({}^{\mathbf{c}}\mathbf{X}) \tag{1.7}$$

Il est possible de retrouver les coordonnées sphériques d'un point à partir de ses coordonnées image puisque l'équation 1.7 est inversible :

$${}^s\mathbf{X} = {pr_s}^{-1}(\mathbf{x}) \quad avec \quad \begin{cases} {}^sX = \dfrac{\xi + \sqrt{1-(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} x \\ {}^sY = \dfrac{\xi + \sqrt{1-(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} y \\ {}^sZ = \dfrac{\xi + \sqrt{1-(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} - \xi \end{cases} . \tag{1.8}$$

Les coordonnées 3D du point dans le repère caméra peuvent également être retrouvées à condition de connaître la profondeur de ce point, c'est-à-dire la distance séparant ce point 3D et le centre optique de la caméra.

Les caméras de type fisheye et catadioptrique sont très prisées en robotique mobile car une grande partie de l'environnement entourant le robot peut être capturée en une seule prise de vue. Néanmoins, ce type de caméra possède des inconvénients. Les images acquises avec ces caméras ont une faible résolution et cette résolution n'est pas uniforme. En effet, plus on s'éloigne du centre d'une image acquise avec une caméra fisheye, plus la résolution est mauvaise. À l'inverse, la résolution des images obtenues avec un système catadioptrique est meilleure aux extrémités et moins bonne au centre de l'image. Enfin, les images provenant d'un dispositif catadioptrique ont en leur centre un angle mort plus ou moins important qui est directement lié à la conception du dispositif. D'autres méthodes permettent d'acquérir des images grand angle avec une meilleure résolution mais au détriment d'un temps d'acquisition, de traitement et de contraintes beaucoup plus importants.

1.1.1.3 Caméra Équirectangulaire

Il est possible d'augmenter le champ de vue d'un dispositif en utilisant plusieurs caméras ou en utilisant les images acquises par une caméra en mouvement.

(a) (b)

FIGURE 1.5: Exemples de dispositif d'acquisition sphérique : (a) système multi-caméras Dodeca 2360 (Immersive Media) et (b) tête rotative motorisée Panogear (Kolor)

La première approche consiste à fusionner les images acquises par un système multi-caméras (ex : Figure 1.5a). Indépendamment les unes des autres, ces images ne couvrent qu'un champ de vue restreint mais il est possible de les assembler pour obtenir un très large champ de vision de type sphérique. L'assemblage se

déroule généralement en trois étapes : l'alignement des images, la déformation des images, et enfin le mélange.

Il est également possible de générer des images équirectangulaires à partir d'une caméra mobile montée sur une tête rotative (ex : Figure 1.5b). La tête rotative pouvant effectuer des rotations horizontales sur 360° et verticales sur 180°, la caméra est orientable dans toutes les directions autour de son centre optique (point nodal). Toutes les positions pouvant être prises par la caméra forment alors une sphère. Lorsqu'une image est projetée sur cette sphère, elle doit parfaitement être juxtaposée géométriquement et photométriquement avec ses images voisines. Assembler un panorama revient à trouver la bonne position de chacune des images sur cette sphère.

La Figure 1.6 illustre les projections permettant d'obtenir une image équirectangulaire. Un point 3D ${}^{\mathbf{c}}\mathbf{X}$ appartenant au repère caméra ${}^{c}\mathcal{R}$ est d'abord projeté sur une sphère unitaire par la projection sphérique $pr_s()$ (eq. 1.5). La position du point ${}^{s}\mathbf{X}$ peut être exprimée selon ses coordonnées polaires de latitude ϕ et de longitude λ dans la sphère par la projection équirectangulaire :

$$\mathbf{x} = pr_e({}^{\mathbf{c}}\mathbf{X}) \quad avec \quad \begin{cases} x = \phi = arctan({}^{s}Z/{}^{s}X) \\ y = \lambda = 2arccos({}^{s}Y) \end{cases} . \tag{1.9}$$

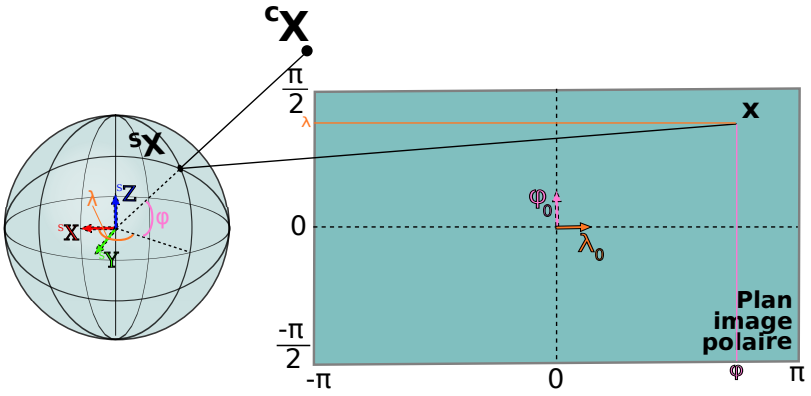


FIGURE 1.6: Schéma de la projection équirectangulaire d'un point du monde 3D dans l'image 2D en passant par la sphère unitaire

Enfin, pour obtenir l'image digitale équirectangulaire, les coordonnées polaires sont transformées en coordonnées cartésiennes à partir de la matrice des paramètres intrinsèques $\mathbf{K}$ (eq. 1.3). Même dans l'image équirectangulaire, les coordonnées de chaque pixel sont directement liées aux coordonnées longitudinales et latitudinales du point 3D dans la sphère. La Figure 1.7 est un exemple d'image équirectangulaire obtenue avec ce type de dispositif.

FIGURE 1.7: Image équirectangulaire de l'intérieur de la cathédrale d'Amiens obtenue avec une caméra montée sur une tête rotative motorisée (E-Cathédrale)

Il est possible de retrouver les coordonnées sphériques d'un point à partir de ses coordonnées image :

$$ {}^{s}\mathbf{X} = {pr_e}^{-1}(\mathbf{x}) \quad avec \quad \begin{cases} {}^{s}X = sin(\frac{\pi(y+1)}{2})cos(-\pi x) \\ {}^{s}Y = cos(\frac{\pi(y+1)}{2}) \\ {}^{s}Z = sin(\frac{\pi(y+1)}{2})sin(-\pi x) \end{cases} . \tag{1.10} $$

Une fois encore, les coordonnées 3D du point dans le repère caméra peuvent être retrouvées à condition de connaître la profondeur de ce point.

Les deux procédés permettant de créer des images équirectangulaires ont des problématiques et des contraintes très proches. Ces approches engendrent des problèmes de positionnement, de synchronisation des prises des vues et de temps d'acquisition des images. En effet, là où il faut une seule acquisition avec une caméra munie d'un objectif fisheye ou catadioptrique, il en faut plusieurs, ou avec un dispositif tournant, pour reconstituer la totalité d'une image. Par conséquent, ces méthodes sont difficilement utilisables dans le cas de scènes dynamiques ou embarquées sur un robot. Cependant, ce type de projection peut s'avérer très utile pour représenter ou naviguer dans un environnement virtuel [Ardouin 2013].

1.1.2 Modélisation d'une scène

Dans la modélisation des différents types de caméra, nous avons représenté les points 3D comme appartenant au repère caméra ${}^{c}\mathcal{R}$. Cependant, les points 3D d'une scène sont généralement représentés dans un repère orthonormé qui est propre à la scène elle-même et que l'on notera ${}^{s}\mathcal{R}$. La figure 1.8 illustre une scène attachée à son repère ${}^{s}\mathcal{R}$ dans laquelle évolue une caméra attachée à son repère ${}^{c}\mathcal{R}$.

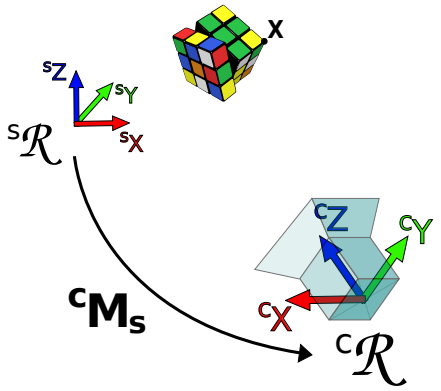


FIGURE 1.8: Représentation d'une scène 3D : relation géométrique entre la scène et une caméra

L'état relatif entre la scène et la caméra dans l'espace peut être décrit à partir de transformations 3D rigides entre leurs repères orthonormés respectifs. Nous notons ${}^{c}\mathbf{R}_{s}$ l'orientation relative entre le repère de la scène et le repère de la caméra. Cette transformation appartient au groupe spécial des matrices orthogonales $\mathbf{SO}(3)$. ${}^{c}\mathbf{t}_{s}$ exprime la position relative entre les deux repères dans $\mathbb{R}^3$. Ces deux transformations composent la matrice homogène de transformation rigide ${}^{c}\mathbf{M}_{s}$ dans le groupe spécial des transformations euclidiennes $\mathbf{SE}(3)$:

$$ {}^{c}\mathbf{M}_{s} = \begin{pmatrix} {}^{c}\mathbf{R}_{s(3\times3)} & {}^{c}\mathbf{t}_{s(1\times3)} \\ \mathbf{0}_{(3\times1)} & 1 \end{pmatrix}_{(4\times4)} \tag{1.11} $$

Ainsi, un point 3D ${}^{s}\mathbf{X} = ({}^{s}X, {}^{s}Y, {}^{s}Z, 1)^{T}$ appartenant au repère de la scène s'exprime dans le repère de la caméra par :

$$ {}^{c}\mathbf{X} = \begin{pmatrix} {}^{c}X \\ {}^{c}Y \\ {}^{c}Z \\ 1 \end{pmatrix} = {}^{c}\mathbf{M}_{s}\ {}^{s}\mathbf{X} \tag{1.12} $$

Inversement, le passage d'un point 3D appartenant au repère caméra vers le repère de la scène se calcule à l'aide de ${}^{c}\mathbf{M}_{s}{}^{-1}$.

Les rotations de la matrice ${}^{c}\mathbf{R}_{s}$ peuvent également être représentées sous différents formalismes :

— Angles d'Euler :
Les angles d'Euler décrivent l'orientation de la scène par rapport à la caméra (ou inversement) sous la forme de trois angles de rotation autour des axes ${}^{s}X$, ${}^{s}Y$ et ${}^{s}Z$ (ou ${}^{c}X$, ${}^{c}Y$ et ${}^{c}Z$) respectivement. Les rotations s'effectuent les unes après les autres, l'ordre dans lequel elles sont réalisées est donc important. Dans la suite, nous noterons ces rotations θ_x, θ_y et θ_z. Une matrice homogène ${}^{c}\mathbf{M}_{s}$ pourra donc être représentée par un vecteur de pose noté $\mathbf{r} = (t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$ où les trois premiers paramètres décrivent la position et les trois derniers sont les angles d'Euler.

— Axe/Angle :
Une rotation dans un espace 3D peut être décrite par deux éléments : un vecteur axial unitaire $\vec{\mathbf{u}}$ indiquant la direction de l'axe de rotation et un angle θ décrivant l'amplitude de la rotation autour de cet axe. L'orientation d'une scène, ou d'une caméra, dans l'espace sous ce formalisme s'écrit alors : $\theta\mathbf{u} = (\theta_{u_x}, \theta_{u_y}, \theta_{u_z})$ avec $\theta = ||\theta\mathbf{u}||$ et $||\vec{\mathbf{u}}|| = 1$.

— Quaternions :
Les quaternions permettent d'étendre à l'espace 3D les propriétés des nombres complexes dans le plan. Un quaternion $\mathbf{q}$ est défini via l'usage de quatre valeurs réelles (s, x, y, z). Ces valeurs sont calculées par une combinaison des trois coordonnées de l'axe de rotation et de l'angle correspondant. En considérant un repère d'axes $\mathbf{i}, \mathbf{j}, \mathbf{k}$, le quaternion représentant une rotation d'angle θ autour du vecteur axial unitaire $\vec{\mathbf{u}}$ s'écrit :

$$\mathbf{q} = s + x\mathbf{i} + \mathbf{j}y + \mathbf{k}z = cos(\frac{\theta}{2}) + \vec{\mathbf{u}}sin(\frac{\theta}{2}) \tag{1.13}$$

En combinant la modélisation de la caméra et la modélisation de son positionnement dans l'espace, il est désormais possible de modéliser le processus complet de génération d'une image 2D d'une scène 3D. Par exemple, les coordonnées images $\mathbf{u}$ d'un point 3D ${}^{s}\mathbf{X}$ appartenant à la scène se trouvent par :

$$\mathbf{u} = \mathbf{K}\, pr_x({}^{c}\mathbf{M}_{s}\, {}^{s}\mathbf{X}) \tag{1.14}$$

avec $pr_x()$ pouvant être la projection perspective (eq. 1.1), la projection omnidirectionnelle (eq. 1.8) ou encore la projection équirectangulaire (eq. 1.9).

1.2 Asservissement visuel

L'asservissement visuel (AV) désigne les méthodes de contrôle en boucle fermée des mouvements d'un système dynamique en utilisant des données visuelles

comme retour d'informations [Weiss 1987, Hutchinson 1996, Chaumette 2006]. Ces méthodes ont tout d'abord été expérimentées sur des robots manipulateurs puis ont été étendues aux robots mobiles.

Les données visuelles peuvent être acquises à partir d'une caméra directement montée sur le robot. Ainsi, les mouvements du système induisent le déplacement de la caméra. On parle dans ce cas de figure d'une configuration "eye-in-hand", on privilégiera cette dernière dans nos travaux (Figure 1.9). La caméra peut être fixée dans l'espace de travail afin d'observer les mouvements du robot à partir d'un point de vue extérieur au système. On parle dans ce cas de figure d'une configuration "eye-to-hand". L'emploi de plusieurs caméras est également possible mais ne fait pas partie des travaux de cette thèse.

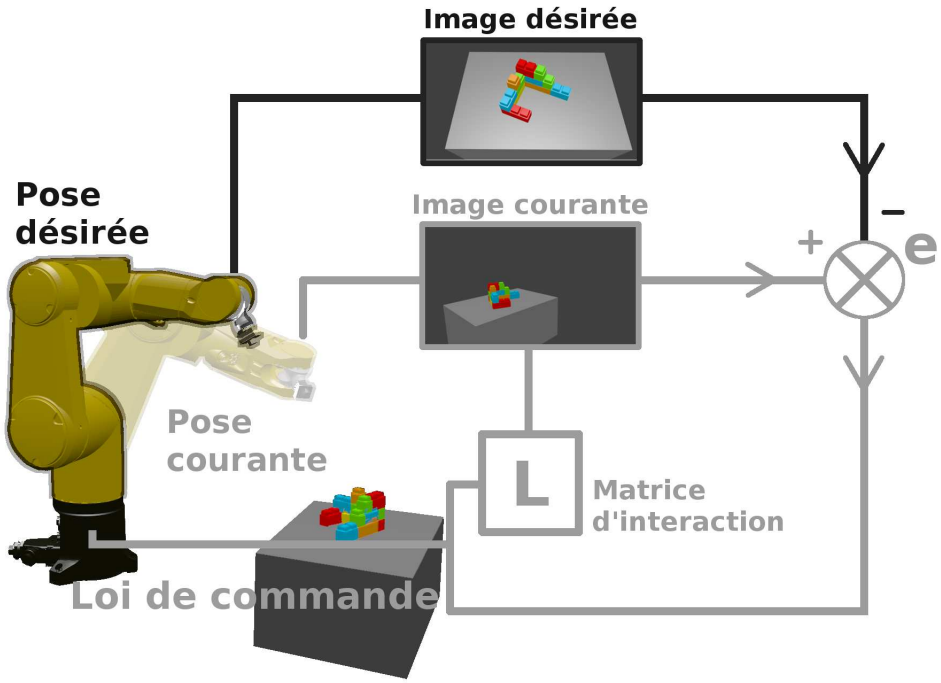


FIGURE 1.9: Schéma d'un système robotisé contrôlé par asservissement visuel en coniguration "eye-in-hand"

En configuration "eye-in-hand", l'AV utilise l'information visuelle perçue par la caméra pour déterminer une loi de commande visant à amener l'effecteur du robot à une pose telle que l'image courante acquise par la caméra et l'image acquise, au préalable, à une pose désirée soient identiques.

Les différentes méthodes d'asservissement visuel peuvent être classées en deux groupes : les méthodes utilisant des primitives géométriques extraites dans les images acquises par la caméra et les méthodes utilisant l'intégralité de l'information photométrique contenue dans les images.

1.2.1 Asservissements visuels basés primitives géométriques

Les approches basées primitives géométriques utilisent des mesures extraites dans l'image désirée et extraites dans les images acquises tout au long de l'asservissement. Ces mesures correspondent généralement à la projection dans l'image 2D de primitives 3D de la scène.

En fonction de la façon dont les primitives 2D extraites des images sont utilisées, les asservissements visuels basés primitives géométriques peuvent être classés en deux sous-catégories :

1. les asservissements visuels basés image
2. les asservissements visuels basés pose

Concernant la première sous-catégorie, les primitives 2D observées dans les images sont directement utilisées dans la tâche de régulation de l'erreur. Les asservissements faisant partie de la seconde sous-catégorie utilisent les primitives 2D afin d'estimer la pose (position et orientation) de la caméra. La régulation de l'erreur s'effectue sur les paramètres 3D de la pose ainsi obtenue. Nous pouvons également citer les asservissements visuels hybrides qui visent à combiner les deux approches afin de ne garder que le meilleur de l'une et de l'autre.

1.2.1.1 Asservissement visuel basé image

Le but de l'asservissement visuel basé image est de déplacer la caméra afin de minimiser l'écart entre les primitives observées dans les images (points [Chaumette 2006], droites [Andreff 2002], régions [Dahmouche 2012], moments [Tahri 2015]...) acquises au cours de l'asservissement et les primitives observées dans l'image désirée. Dans ce cas, l'erreur $\mathbf{e}$ que l'on cherche à réguler peut s'écrire :

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

où $\mathbf{s}(\mathbf{r})$ contient les primitives extraites de l'image acquise par la caméra à la pose $\mathbf{r} = (t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$ et $\mathbf{s}^*$ contient les primitives, considérées comme constantes, extraites de l'image désirée.

Notons $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{w})$ le torseur cinématique de la pose de la caméra composé de $\boldsymbol{v} = (v_X, v_Y, v_Z)$ la vitesse de translation et $\boldsymbol{w} = (w_X, w_Y, w_Z)$, la vitesse de rotation. Les déplacements d'une primitive $\mathbf{s}$ dans l'image sont liés à la vitesse de déplacement $\mathbf{v}$ de la caméra [Espiau 1992] :

$$\dot{\mathbf{s}} = \frac{d\mathbf{s}}{dt} = \frac{\partial \mathbf{s}}{\partial \mathbf{r}} \frac{d\mathbf{r}}{dt} = \frac{\partial \mathbf{s}}{\partial \mathbf{S}} \frac{\partial \mathbf{S}}{\partial \mathbf{r}} \frac{d\mathbf{r}}{dt} = \mathbf{L_s v} \tag{1.16}$$

La matrice $\mathbf{L_s}$ reliant les déplacements des primitives de l'image aux mouvements de la caméra est appelée matrice d'interaction [Chaumette 2006]. Comme

le montre l'équation (1.16), la matrice d'interaction est composée de deux jacobiens. Le jacobien liant les primitives de l'image aux primitives 3D $\mathbf{S}$ exprimées dans le repère caméra dépend directement du modèle de projection de la caméra. Les matrices d'interaction pour un grand nombre de primitives (points, droites, sphères, moments...) ont d'abord vu le jour en vision perspective [Feddema 1989, Chaumette 1990]. Puis, grâce au modèle de projection unifié [Barreto 2001], ces matrices d'interaction ont été généralisées à la vision omnidirectionnelle [Barreto 2004, Tahri 2010, Hadj-Abdelkader 2010].

À partir des équations (1.15) et (1.16), il est possible de relier la variation temporelle de l'erreur aux déplacements de la caméra :

$$\dot{\mathbf{e}} = \mathbf{L_s}\mathbf{v} \tag{1.17}$$

Pour s'assurer autant que possible d'une décroissance exponentielle de l'erreur ($\dot{\mathbf{e}} = -\lambda\mathbf{e}$), la loi de contrôle de l'asservissement visuel devient :

$$\mathbf{v} = -\lambda\mathbf{L_s}^{+}\mathbf{e} \tag{1.18}$$

où le gain λ permet de régler la vitesse de convergence et où $\mathbf{L_s}^{+}$ est la pseudo-inverse de $\mathbf{L_s}$. Même si la matrice d'interaction $\mathbf{L_s}$ possède une forme analytiquement connue, certains des paramètres utilisés dans son calcul sont approximatifs (paramètres intrinsèques de la caméra, profondeur des points de la scène). Par conséquent, en pratique nous utilisons une approximation de la matrice d'interaction que l'on notera $\widehat{\mathbf{L_s}}$. La loi de commande devient alors :

$$\mathbf{v} = -\lambda\widehat{\mathbf{L_s}}^{+}\mathbf{e} \tag{1.19}$$

La stabilité d'un système commandé par asservissement visuel peut être étudiée à partir de la notion de stabilité de Lyapunov. Au sens de Lyapunov, un système dynamique décrit par l'équation différentielle $\dot{\mathbf{x}} = f(\mathbf{x}, t)$ est considéré comme stable en un point d'équilibre $\mathbf{x_e}$ si et seulement s'il existe une fonction L (dite fonction de Lyapunov) vérifiant certaines conditions précises en lien direct avec la fonction f et $\mathbf{x_e}$. L'état $\mathbf{x_e}$ est un point d'équilibre du système si $L > 0$ pour tout $\mathbf{x}$ et si $L(\mathbf{x_e}) = 0$.

La stabilité d'un système commandé par asservissement visuel peut être étudiée à partir de la fonction de Lyapunov suivante :

$$L = \frac{1}{2}||\mathbf{e(t)}||^2 \tag{1.20}$$

La dérivée de L par rapport au temps, nous donne :

$$\dot{L} = \mathbf{e}^T\dot{\mathbf{e}} \tag{1.21}$$

À partir des équations (1.17) et (1.19), nous obtenons :

$$\dot{L} = -\lambda \mathbf{e}^T \mathbf{L_s} \widehat{\mathbf{L_s}}^+ \mathbf{e} \tag{1.22}$$

D'après le théorème de Lyapunov, la stabilité asymptotique globale du système est assurée si la condition suivante est respectée :

$$\mathbf{L_s} \widehat{\mathbf{L_s}}^+ > 0 \tag{1.23}$$

En d'autres termes, si le nombre de primitives est égal au nombre de degrés de liberté du robot et si $\mathbf{L_s}$ et $\widehat{\mathbf{L_s}}^+$ sont de rang 6 alors la condition (1.23) est assurée. Cependant, la plupart du temps, le nombre de primitives visuelles est supérieur au nombre de degrés de libertés du robot, $\mathbf{L_s}\widehat{\mathbf{L_s}}^+$ est alors au maximum de rang 6. Cela signifie que $\mathbf{s} - \mathbf{s}^*$ appartient à l'espace nul de $\widehat{\mathbf{L_s}}^+$:

$$\mathbf{s} - \mathbf{s}^* = \mathbf{e} \in Ker(\widehat{\mathbf{L_s}}^+) \tag{1.24}$$

Cette configuration correspond à un minimum local de la fonction de coût. Par conséquent, lorsque le nombre de primitives visuelles est supérieur au nombre de degrés de libertés du robot, une stabilité asymptotique globale ne peut être assurée mais il est possible de démontrer qu'il existe une stabilité asymptotique locale.

1.2.1.2 Asservissement visuel basé pose

La régulation de l'erreur en asservissement visuel basé pose est directement liée à la pose de la caméra par rapport à la scène observée [Wilson 1996]. L'estimation de cette pose à partir de primitives extraites des images acquises par la caméra nécessite la connaissance des paramètres intrinsèques de la caméra ainsi qu'un modèle 3D de la scène.

La pose de la caméra peut être représentée selon différents formalismes (angles d'Euler, quaternions, axe/angle...). Le vecteur $\mathbf{s}^*$ de l'équation (1.15) contient les paramètres de représentation de la pose calculée à partir des primitives observées dans l'image désirée. Le vecteur $\mathbf{s}(\mathbf{r})$ contient les paramètres de représentation de la pose calculée à partir des primitives observées dans l'image acquise à partir de la pose $\mathbf{r}$.

La loi de contrôle peut être exprimée par rapport à la scène ou par rapport à la pose désirée de la caméra [Chaumette 2007]. En considérant le formalisme axe/angle pour représenter la pose de la caméra (pour ${}^{c^*}R_c$) et en exprimant la loi de contrôle par rapport à la scène, nous obtenons $\mathbf{s}(\mathbf{r}) = ({}^c\mathbf{t}_s, \theta\mathbf{u})$, $\mathbf{s}^* = ({}^{c^*}\mathbf{t}_s, 0)$ et $\mathbf{e} = ({}^c\mathbf{t}_s - {}^{c^*}\mathbf{t}_s, \theta\mathbf{u})$. Dans ces conditions, la matrice d'interaction est :

$$\mathbf{L_s} = \begin{pmatrix} -\mathbf{I}_3 & [{}^c\mathbf{t}_s]_\times \\ \mathbf{0} & \mathbf{L}_{\theta\mathbf{u}} \end{pmatrix} \tag{1.25}$$

où $\mathbf{I}_3$ est une matrice identité de taille 3×3, $[{}^c\mathbf{t}_s]_\times$ l'antisymétrique de ${}^c\mathbf{t}_s$ et où $\mathbf{L}_{\theta\mathbf{u}}$ est calculée par :

$$\mathbf{L}_{\theta\mathbf{u}} = \mathbf{I}_3 + \frac{\theta}{2}[\mathbf{u}]_\times + (1 + \frac{sinc\,\theta}{sinc^2\,\frac{\theta}{2}})[\mathbf{u}]_\times^2 \tag{1.26}$$

avec *sinc* correspondant à la fonction sinus cardinal et $[\mathbf{u}]_\times$ la matrice antisymétrique de $\mathbf{u}$.

En partant de la même loi de commande que précédemment (eq. (1.19)), le torseur cinématique de la pose de la caméra devient :

$$\mathbf{v} = \begin{cases} \boldsymbol{v} = -\lambda(({}^{c^*}\mathbf{t}_s - {}^c\,\mathbf{t}_s) - [{}^c\mathbf{t}_s]_\times \theta\mathbf{u}) \\ \boldsymbol{w} = -\lambda\theta\mathbf{u} \end{cases} \tag{1.27}$$

Sous cette configuration, les rotations de la caméra suivent une géodésique avec une vitesse décroissante exponentiellement. Dans l'image, le centre du repère de la scène suit une ligne droite. Par conséquent, la trajectoire spatiale de la caméra est incurvée et non rectiligne.

La seconde configuration permet de découpler les rotations et les translations de la caméra. En exprimant la loi de contrôle par rapport à la pose désirée de la caméra, nous obtenons $\mathbf{s}(\mathbf{r}) = ({}^{c*}\mathbf{t}_c, \theta\mathbf{u})$, $\mathbf{s}^* = \mathbf{0}$ et $\mathbf{e} = \mathbf{s}$. Dans ces conditions, la matrice d'interaction est :

$$\mathbf{L}_\mathbf{s} = \begin{pmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{\theta\mathbf{u}} \end{pmatrix} \tag{1.28}$$

et le torseur cinématique de la caméra devient :

$$\mathbf{v} = \begin{cases} \boldsymbol{v} = -\lambda\mathbf{R}^{T\,c*}\mathbf{t}_c \\ \boldsymbol{w} = -\lambda\theta\mathbf{u} \end{cases} \tag{1.29}$$

Sous cette configuration, la trajectoire spatiale de la caméra est rectiligne. Cependant les images acquises pendant l'asservissement peuvent plus facilement conduire à la sortie de primitives visuelles du champ de vue de la caméra et par conséquent à l'échec de l'asservissement.

Que ce soit basé image ou basé pose, le bon fonctionnement des asservissements visuels basés primitives visuelles dépend principalement des mesures extraites dans les images. Ces extractions utilisent des techniques de traitement d'image de détection de primitives visuelles, de mise en correspondance, de suivi ou encore de segmentation. Bien qu'ayant été, et étant toujours, très étudiées, ces techniques restent très délicates. C'est pourquoi, des approches n'ayant pas besoin de passer par ces traitements d'image ont été proposées.

1.2.2 Asservissements visuels photométriques

Plutôt que de représenter une scène par un ensemble de caractéristiques géométriques extraites d'une image, il a été proposé d'utiliser directement l'apparence de la scène, c'est-à-dire d'utiliser l'image dans son ensemble [Collewet 2008]. Cette démarche permet de passer outre aux problèmes liés à la détection, la mise en correspondance, le suivi ou encore la segmentation de l'image. Qui plus est, l'utilisation de l'image complète apporte une redondance d'information visuelle, ce qui est doublement intéressant en vison omnidirectionnelle [Caron 2010a].

Dans un premier temps, il a été proposé de travailler dans des sous-espaces de dimensions réduites [Nayar 1996, Deguchi 2000]. Par exemple, une Analyse en Composantes Principales (ACP) permet de décomposer l'image en valeurs propres. La commande est ensuite réalisée dans l'espace des valeurs propres et la matrice d'interaction liée à l'espace propre est apprise hors ligne. Cet asservissement se base donc sur un apprentissage en aval et non sur une modélisation analytique des caractéristiques de l'image ou de la matrice d'interaction.

Certains travaux [Benhimane 2004, Benhimane 2007] se placent à mi-chemin entre les asservissements visuels basés primitives géométriques et les asservissements visuels photométriques. En effet, ces asservissements ne nécessitent pas d'extraire des caractéristiques visuelles dans les images. La fonction de coût à minimiser se base sur la transformation linéaire, si ce n'est l'homographie, entre un plan de la scène visible dans l'image désirée et ce même plan dans les images courantes.

Plus récemment, des asservissements visuels dans lesquels toutes les intensités contenues dans les images sont directement utilisées en entrée de la loi de commande [Collewet 2008] ont été développés. Ce sont les asservissements visuels photométriques.

1.2.2.1 Asservissement visuel purement photométrique

Nous pouvons définir un asservissement visuel comme purement photométrique, lorsque la fonction d'erreur est directement définie comme étant la différence entre les intensités de l'image désirée et les intensités des images acquises par la caméra au cours de l'asservissement. La fonction d'erreur (eq.(1.15)) peut alors s'écrire :

$$\mathbf{e} = \mathbf{I}(\mathbf{r}) - \mathbf{I}^* \tag{1.30}$$

où $\mathbf{I}(\mathbf{r})$ est un vecteur de taille $N \times M$ contenant l'intensité des pixels de l'image $\mathbf{I}$ de taille $N \times M$ acquise à la pose de caméra $\mathbf{r}$. $\mathbf{I}^*$ est un vecteur de taille $N \times M$ contenant l'intensité des pixels de l'image acquise à la pose désirée.

Comme pour les asservissements visuels présentés précédemment, afin d'établir une loi de commande, il est nécessaire de calculer la matrice d'interaction

liée aux intensités de l'image. En considérant que l'intensité I d'un même point $\mathbf{x}$ reste constante après un mouvement de la caméra, on peut écrire :

$$I(\mathbf{x} + d\mathbf{x}, t + dt) = I(\mathbf{x}, t) \tag{1.31}$$

avec $d\mathbf{x}$ le déplacement du point $\mathbf{x}$ dans l'image et dt l'intervalle de temps entre les deux images. Si $d\mathbf{x}$ est faible alors la contrainte du flot optique [Horn 1980] est valide :

$$\nabla I^T \dot{\mathbf{x}} + I_t = 0 \tag{1.32}$$

avec $\vec{\nabla} I$ le gradient spatial de $I(\mathbf{x}, t)$ et $I_t = \frac{\partial I(\mathbf{x},t)}{\partial t}$ le gradient temporel.

En reprenant l'équation 1.16 pour une primitive de type point, les déplacements d'un point dans l'image sont liés aux mouvements de la caméra par : $\dot{\mathbf{x}} = \mathbf{L_x v}$ avec $\mathbf{L_x}$ la matrice d'interaction associée aux points. L'équation 1.32 peut alors devenir :

$$\nabla I^T \mathbf{L_x v} + I_t = 0 \tag{1.33}$$

La matrice d'interaction $\mathbf{L}_I(\mathbf{x})$ reliant l'intensité du point $\mathbf{x}$ aux mouvements de la caméra peut alors être obtenue sous la forme :

$$\mathbf{L}_I(\mathbf{x}) = -\nabla I^T \mathbf{L_x} \tag{1.34}$$

La matrice d'interaction $\mathbf{L_I}(\mathbf{x})$ se base sur la contrainte du flot optique. Elle ne devrait être valable que pour une scène Lambertienne éclairée par une source de lumière immobile. Cependant, les nombreuses utilisations pratiques de ce formalisme ont montré que l'approche est fonctionnelle même lorsque les hypothèses de base ne sont pas remplies.

Le gradient spatial ∇I est la seule information obtenue par traitement d'image. Dans le cas perspectif, les gradients image sont généralement calculés à l'aide de deux filtres dérivatifs de taille fixe le long des axes $\vec{\mathbf{x}}$ et $\vec{\mathbf{y}}$ de l'image dont les coefficients correspondent aux dérivées d'un filtre gaussien :

$$\mathbf{F_x} = \frac{1}{8418}(-112, -913, -2047, 0, 2047, 913, 112) \tag{1.35}$$

$$\mathbf{F_y} = \mathbf{F_x}^T \tag{1.36}$$

Dans le cas omnidirectionnel, la résolution et l'orientation dans l'image ne sont pas constantes. C'est pourquoi, il a été proposé d'adapter la forme des filtres utilisés pour le calcul des gradients [Caron 2010b]. Le voisinage d'un point de l'image est d'abord projeté sur un pôle de la sphère puis est déplacé afin d'être centré aux coordonnées sphériques de ce point. Pour finir, le voisinage est projeté sur le plan image et les intensités des pixels sont obtenues par interpolation.

En considérant la pose désirée que l'on souhaite atteindre comme étant la solution d'un problème d'optimisation non-linéaire, différentes méthodes d'optimisations peuvent être utilisées pour déterminer la loi de commande de la caméra.

Par exemple, une simple méthode de descente du gradient de la fonction de coût peut être employée, la loi de commande obtenue est alors :

$$\mathbf{v} = -\lambda {\mathbf{L_I}}^T(\mathbf{I}(\mathbf{r}) - \mathbf{I}^*) \tag{1.37}$$

avec $\mathbf{L_I}$ la matrice d'interaction associée à l'intensité des pixels de l'image $\mathbf{I}$. $\mathbf{L_I}$ contient les matrices d'interactions (eq. 1.34) de chaque pixel de l'image $\mathbf{I}$ (de taille $N \times M$) empilées sous la forme :

$$\mathbf{L_I} = \begin{pmatrix} \mathbf{L}_I(0) \\ \mathbf{L}_I(1) \\ \mathbf{L}_I(2) \\ \dots \\ \mathbf{L}_I(N \times M) \end{pmatrix} \tag{1.38}$$

La méthode de Gauss-Newton peut également être utilisée, la loi de commande devient alors :

$$\mathbf{v} = -\lambda {\mathbf{L_I}}^+(\mathbf{I}(\mathbf{r}) - \mathbf{I}^*) \tag{1.39}$$

Enfin, il a été démontré que la loi de commande la plus efficace est obtenue en se basant sur la méthode de Levenberg-Marquadt [Collewet 2008] :

$$\mathbf{v} = -\lambda(\mathbf{H} + \mu\ diag(\mathbf{H}))^{-1}{\mathbf{L_I}}^T(\mathbf{I}(\mathbf{r}) - \mathbf{I}^*) \tag{1.40}$$

avec $\mathbf{H} = {\mathbf{L_I}}^T\mathbf{L_I}$. Cette loi de commande assure une meilleure convergence car elle se comporte comme une descente de gradient (eq. 1.37) lorsque μ est élevée et comme un Gauss-Newton (eq. 1.39) lorsque μ est faible. Il faut donc déterminer μ judicieusement.

L'utilisation directe de l'intégralité de l'information visuelle contenue dans les images pour l'asservissement visuel s'avère être très efficace, et ce avec comme unique traitement d'image le calcul des gradients de l'image. Même si la scène ne respecte pas les conditions Lambertiennes imposées dans le calcul de la matrice d'interaction, l'erreur de positionnement à convergence reste très faible. La méthode est également fiable même avec d'importantes approximations sur les profondeurs. Cependant, la méthode se montre peu robuste aux changements d'illuminations et aux occultations dans la scène. Enfin, les trajectoires empruntées par la caméra au cours des asservissements peuvent être très chaotiques. Le nombre de primitives visuelles étant largement supérieur au nombre de degrés de liberté du robot, comme pour l'asservissement visuel basé primitives géométriques dans cette configuration, nous ne pouvons assurer qu'une stabilité asymptotique locale et non globale.

Différentes améliorations ont été proposées pour résoudre les problèmes liés aux variations d'illumination et aux occultations. L'approche [Comport 2003b]

propose d'utiliser un M-Estimateur issu des statistiques robustes [Huber 1981]. Le principe consiste à plus ou moins pondérer les valeurs d'erreur entre les pixels de l'image courante et de l'image désirée en se basant sur l'écart type des erreurs. Dans [Delabarre 2012], les auteurs ont proposé d'adapter les intensités de l'image désirée en fonction de celles de l'image courante tout au long de l'asservissement. Enfin, d'autres méthodes proposent également d'utiliser l'intégralité des pixels contenus dans les images mais de manière indirecte. En effet, il n'est plus question de minimiser directement l'erreur entre l'image désirée et les images acquises pendant l'asservissement mais d'utiliser différemment l'information photométrique.

1.2.2.2 Asservissement visuel basé information mutuelle

Contrairement au type d'asservissement présenté précédemment, l'asservissement visuel basé information mutuelle [Dame 2010] ne cherche pas à directement minimiser l'erreur photométrique mais à maximiser une mesure de similarité calculée entre l'image désirée et les images acquises à chaque itération de l'optimisation. L'information mutuelle [Shannon 1948, Viola 1997] correspond à la quantité d'informations photométriques communes aux deux images. Cette mesure de similarité s'est montrée robuste aux bruits, aux réflexions spéculaires et s'avère être intéressante pour comparer deux images ayant des modalités différentes [Dame 2012].

L'information mutuelle IM d'une image désirée $\mathbf{I}^*$ et d'une image $\mathbf{I}(\mathbf{r})$ est calculée à partir de l'entropie des deux images, respectivement $H(\mathbf{I}^*)$ et $H(\mathbf{I}(\mathbf{r}))$, et de leur entropie jointe $H(\mathbf{I}^*, \mathbf{I}(\mathbf{r}))$:

$$IM(\mathbf{I}^*, \mathbf{I}(\mathbf{r})) = H(\mathbf{I}^*) + H(\mathbf{I}(\mathbf{r})) - H(\mathbf{I}^*, \mathbf{I}(\mathbf{r})) \tag{1.41}$$

L'entropie, au sens de Shannon [Shannon 1948], de l'image $\mathbf{I}^*$ mesure la variabilité des intensités $I \in [0, 255]$ la composant :

$$H(\mathbf{I}^*) = -\sum_{I=0}^{255} p_{\mathbf{I}^*(I)} \log(p_{\mathbf{I}^*(I)}) \tag{1.42}$$

avec $p_{\mathbf{I}^*(I)}$ la probabilité qu'un pixel de l'image $\mathbf{I}^*$ ait l'intensité I. $p_{\mathbf{I}^*}$ n'est autre que l'histogramme normalisé de l'image $\mathbf{I}^*$ décrit par :

$$p_{\mathbf{I}^*(I)} = \frac{1}{N_b} \sum_{\mathbf{x}} \delta(I(\mathbf{x}) - I) \tag{1.43}$$

où δ est la fonction de Kronecker.

L'entropie jointe $H(\mathbf{I}^*, \mathbf{I}(\mathbf{r}))$ mesure la variabilité des intensités $I_1 \in [0, 255]$ et $I_2 \in [0, 255]$ communes aux images $\mathbf{I}^*$ et $\mathbf{I}(\mathbf{r})$:

$$H(\mathbf{I}^*, \mathbf{I}(\mathbf{r})) = -\sum_{I_1=0}^{255} \sum_{I_2=0}^{255} p_{\mathbf{I}^*\mathbf{I}(\mathbf{r})(I_1,I_2)} \log(p_{\mathbf{I}^*\mathbf{I}(\mathbf{r})(I_1,I_2)}) \tag{1.44}$$

avec $p_{\mathbf{I}^*\mathbf{I}(\mathbf{r})(I_1,I_2)}$ la probabilité qu'un pixel de l'image $\mathbf{I}^*$ ait l'intensité I_1 et que ce pixel dans l'image $\mathbf{I}(\mathbf{r})$ ait l'intensité I_2. $p_{\mathbf{I}^*\mathbf{I}(\mathbf{r})}$ est l'histogramme joint des images $\mathbf{I}^*$ et $\mathbf{I}(\mathbf{r})$. L'entropie jointe indique donc la quantité d'informations visuelles apportée simultanément par les deux images. Plus cette valeur est faible, plus les images contiennent d'informations similaires.

Par conséquent, maximiser l'information mutuelle (eq. 1.41) entre deux images revient à chercher à aligner le plus précisément possible ces deux images. Une approximation de Taylor au premier ordre de la fonction IM permet de lier la similarité entre une image désirée et une image acquise à la pose $\mathbf{r}_k$ et la similarité entre l'image désirée et une image acquise à l'itération suivante, à la pose $\mathbf{r}_{k+1}$:

$$IM(\mathbf{r}_{k+1}) \approx IM(\mathbf{r}_k) + \mathbf{L}_{IM(\mathbf{r_k})}\dot{\mathbf{r}}\Delta t \tag{1.45}$$

où $\mathbf{L}_{IM}$ est le vecteur d'interaction reliant la variation de la fonction IM par rapport aux déplacements de la caméra. Au second ordre, nous avons :

$$\mathbf{L}_{IM(\mathbf{r}_{k+1})} \approx \mathbf{L}_{IM(\mathbf{r}_k)} + \mathbf{H}_{IM(\mathbf{r_k})}\dot{\mathbf{r}}\Delta t \tag{1.46}$$

où $\mathbf{H}_{IM(\mathbf{r_k})}$ est la matrice d'interaction de $\mathbf{L}_{IM}$, également appelée la hessienne.

Maximiser l'information mutuelle est similaire à la régulation à 0 de la variation de la fonction IM par rapport aux déplacements de la caméra. En fixant $\Delta t = 1$, l'incrément de pose menant à une variation nulle de l'information mutuelle devient :

$$\dot{\mathbf{r}} = \mathbf{v} = -\mathbf{H}^{-1}_{IM(\mathbf{r_k})}\mathbf{L}^T_{IM(\mathbf{r}_k)} \tag{1.47}$$

Le calcul non-approximé de la hessienne $\mathbf{H}_{IM}$ nécessite le calcul de la double dérivée de la fonction de Kronecker δ. Cela est rendu possible en interpolant les probabilités intervenant dans les équations (1.42) et (1.44) par des fonctions B-splines ($\delta = \phi$). Cette formulation ainsi que la réduction du nombre de classes des histogrammes permettent de lisser la forme de la fonction de coût de l'IM.

1.2.2.3 Asservissement visuel basé histogrammes d'intensité

Plus récemment, [Bateux 2015] ont proposé de représenter les images sous la forme de plusieurs histogrammes d'intensité. L'histogramme d'une image peut être considéré comme une représentation globale de son information visuelle. C'est pourquoi ce descripteur, qu'il soit d'intensités ou non, est largement utilisé en vision par ordinateur, que ce soit pour la détection et la mise en correspondance ou encore le suivi [Dalal 2005, Lowe 2004].

Comme présenté précédemment, l'histogramme d'une image est une fonction qui associe à chaque valeur d'intensité le nombre de pixels dans l'image ayant cette valeur (eq 1.43). Cette fois, l'erreur à minimiser est la distance entre l'histogramme de l'image désirée et l'histogramme de l'image courante. La distance

de Matusita [Cha 2002] est utilisée pour comparer ces deux histogrammes :

$$\rho(\mathbf{I}^*, \mathbf{I}(\mathbf{r})) = \frac{1}{N_b} \sum_{i}^{N_b} (\sqrt{\mathbf{p_{I(r)}}} - \sqrt{\mathbf{p_I}})^2 \tag{1.48}$$

avec N_b le nombre de classes des histogrammes.

Comme précédemment, la fonction de Kronecker δ (eq. 1.43) utilisée dans le calcul des histogrammes n'étant pas différentiable, cette fonction est remplacée par une fonction B-splines d'ordre 2.

Après plusieurs manipulations et simplifications, la matrice d'interaction reliant les variations des distances de Matusita entre les deux histogrammes et les déplacements de la caméra peut s'écrire [Bateux 2015] :

$$\mathbf{L}_\rho = \delta\rho(\mathbf{I}^*, \mathbf{I}(\mathbf{r})) = \sum_{1}^{N_b} \left(\frac{i}{N_x} (1 - \frac{\sqrt{\mathbf{p_{I^*(i)}}}}{\sqrt{\mathbf{p_{I(r)(i)}}}}) \right) \sum_{\mathbf{x}}^{N_x} \left(\frac{\delta}{\delta i} \phi(\mathbf{I}(\mathbf{r}, \mathbf{x}) - i)) \mathbf{L_I} \right) \tag{1.49}$$

où $\phi()$ est la fonction B-splines d'ordre 2 remplaçant la fonction de Kronecker δ. La matrice d'interaction $\mathbf{L_I}$ relie les intensités de l'image $\mathbf{I}(\mathbf{r})$ par rapport aux déplacements de la caméra (eq. 1.38).

Afin de rendre l'approche plus robuste aux entrées et aux sorties d'informations visuelles du champ de vue de la caméra, un noyau de pondération est utilisé pour donner plus d'importance aux pixels au centre de l'image.

L'histogramme d'une image étant invariant en rotation, l'utilisation d'un seul histogramme pour représenter chaque image n'est pas suffisante pour contrôler les six degrés de liberté de la caméra. C'est pourquoi, [Bateux 2015] proposent de diviser les images en plusieurs parties et de calculer un histogramme distinct pour chaque partie. La matrice d'interaction correspond alors à l'empilement des matrices d'interaction calculées pour chaque partie indépendamment les unes des autres :

$$\mathbf{L}_\rho = [\mathbf{L}_{\rho\mathbf{1}}^\mathbf{T} \; \mathbf{L}_{\rho\mathbf{2}}^\mathbf{T} \; ... \; \mathbf{L}_{\rho\mathbf{n}}^\mathbf{T}]^T \tag{1.50}$$

où $\mathbf{L_{\rho i}}$ est la matrice d'interaction de la partie i de l'image. D'après les différentes expérimentations menées par les auteurs, découper l'image en vingt-cinq parties représente un bon compromis pour être en mesure de contrôler les six degrés de liberté de la caméra pendant l'asservissement.

1.2.2.4 Asservissement visuel basé noyaux

En se basant sur les méthodes de suivi basé noyaux, [Kallem 2007] ont proposé une méthode d'asservissement visuel ayant pour but de ne pas séparer le suivi de l'information visuelle contenue dans les images et la commande de la caméra. A la différence des méthodes présentées jusque maintenant, plutôt que de

considérer les pixels comme des variables discrètes sur une image finie mesurées en temps discret, les pixels sont considérés comme des variables continues définies sur l'ensemble $\mathbb{R}^2$ mesurées en temps continu. Par conséquent, les images et les transformations qui leur sont apportées sont considérées comme des signaux qui sont directement mesurés.

Un noyau $K : \mathbb{R}^2 \to \mathbb{R}$ est une fonction continue qui est projetée sur l'image. Le noyau agit comme une fonction de pondération et la somme de ces pondérations correspond à ce qui est appelé la valeur de la projection du noyau. Notons $I(\mathbf{x}, t)$, le signal correspondant aux intensités d'une image au fil du temps. Alors la valeur de la projection du noyau K sur ce signal à l'instant t est :

$$\upsilon(t) = \iint_I K(\mathbf{u}) I(\mathbf{x}, t) dxdy \tag{1.51}$$

où $\mathbf{x} = (x, y) \in I = \mathbb{R}^2$ est la variable d'indexation spatiale de l'image.

Cette fois, nous cherchons donc à minimiser l'erreur entre la valeur de la projection du noyau sur le signal de l'image désirée et la valeur de la projection du noyau sur le signal de l'image acquise à la pose courante $\mathbf{r}$:

$$e = \upsilon - \upsilon^* \tag{1.52}$$

En appliquant le théorème de dérivation des fonctions composées sur la fonction de Lyapunov : $V = \frac{1}{2}(\upsilon - \upsilon^*)^2$, nous obtenons :

$$\dot{V} = (\upsilon - \upsilon^*)\dot{\upsilon} = (\upsilon - \upsilon^*)\frac{\partial \upsilon}{\partial \mathbf{r}}\dot{\mathbf{r}} \tag{1.53}$$

où $\frac{\partial \upsilon}{\partial \mathbf{r}}$ est la matrice d'interaction reliant les variations de la valeur de la projection du noyau par rapport aux mouvements de la caméra. Pour que l'équation 1.51 soit différentiable, la fonction noyau $K(\mathbf{x})$ doit être lisse.

Pour satisfaire la condition de stabilité, la dérivée de la fonction de Lyapunov doit être strictement négative : $\dot{V} < 0$. L'entrée de la loi de contrôle peut alors s'écrire :

$$\dot{\mathbf{r}} = \mathbf{v} = -(\upsilon - \upsilon^*) \iint_I \nabla K(\mathbf{x}) F(I(\mathbf{x}, \mathbf{r})) dxdy \tag{1.54}$$

où $\nabla K(\mathbf{x}) = (\frac{\partial K}{\partial \mathbf{x}})^T$ et où $F(I(\mathbf{x}, \mathbf{r}))$ correspond au signal $I(\mathbf{x}, \mathbf{r})$ de l'image acquise à la position $\mathbf{r}$ ayant subi une transformation. Cette transformation est directement liée aux degrés de liberté que l'on souhaite contrôler durant l'asservissement. Dans [Kallem 2007], trois types de mouvements de caméra ont été traités :

— Translations 2D planaires : le signal de l'image correspond directement au signal de l'image sans transformation, autrement dit $F(I(\mathbf{x}, \mathbf{r})) = I(\mathbf{x}, \mathbf{r})$. Le noyau K utilisé est un noyau gaussien 2D pondérant chaque pixel le long des axes $\vec{x}$ et $\vec{y}$ de l'image.

— Translation le long de l'axe optique : ce mouvement de caméra est considéré comme une différence d'échelle entre l'image désirée et l'image courante. Pour contrôler ce mouvement de caméra, l'amplitude de la transformée de Fourrier de l'image est utilisée comme signal. Le noyau K utilisé pour ce mouvement de caméra est un noyau gaussien 1D.
— Rotations autour de l'axe optique : l'amplitude de la transformée de Fourrier étant invariante aux translations planaires, comme pour la mise à l'échelle, cette transformation de l'image est utilisée pour contrôler les rotations autour de l'axe optique de la caméra. Le noyau K utilisé est un noyau asymétrique en rotation : $K(\mathbf{x}) = (r_{max} - (u^2 + v^2))sin^2(\theta)$ où r_{max} est un rayon fixé par l'utilisateur et où θ est l'angle d'orientation du pixel aux coordonnées $\mathbf{x} = (x, y)$.

1.2.2.5 Asservissement visuel basé moments photométriques

L'asservissement visuel basé moments photométriques [Bakthavatchalam 2013] propose d'utiliser indirectement les intensités des images, dans le sens où ces intensités servent à calculer des caractéristiques globales représentant les images. Ces caractéristiques sont appelées moments photométriques. Les moments se sont déjà révélés être des primitives très intéressantes en asservissement visuel basé image (Section 1.2.1.1) [Chaumette 2004, Wang 2008], et ce grâce à leurs propriétés de découplage. Cependant, ces méthodes nécessitent des étapes de détection et de mise en correspondance ou de segmentation des images en régions homogènes pour obtenir de bons résultats d'asservissement. L'utilisation des moments photométriques permet de passer outre à ces étapes.

L'expression générale des moments photométriques dans le plan image I peut s'écrire :

$$m_{pq} = \iint_I x^p y^p I((x, y), t) dx dy \tag{1.55}$$

où $p + q$ définit l'ordre du moment et $I((x, y), t)$ est la fonction d'intensité de l'image. Nous pouvons voir que l'expression des moments photométriques et la définition de la projection d'un noyau sur le signal image (eq. 1.51) sont similaires. Les moments photométriques peuvent être vus comme un cas particulier de la méthode basée noyaux [Kallem 2007].

Comme précédemment, le calcul de la matrice d'interaction reliant les variations des moments photométriques aux mouvements de la caméra est nécessaire. En se basant sur l'hypothèse d'illumination constante dans le temps et en suivant le même raisonnement que pour l'asservissement visuel purement photométrique,

la matrice d'interaction associée aux moments photométriques peut s'écrire :

$$\mathbf{L}_{m_{p,q}} = \iint_I x^p y^p \mathbf{L_I} dxdy \tag{1.56}$$

avec $\mathbf{L_I}$ la matrice d'interaction liant les variations des intensités des pixels de l'image aux mouvements de la caméra (eq. 1.38).

A l'aide du théorème de Green et de plusieurs manipulations mathématiques, la forme finale de la matrice d'interaction $\mathbf{L}_{m_{p,q}}$ devient pour une scène considérée comme étant planaire :

$$\mathbf{L}^T_{m_{p,q}} = \begin{pmatrix} -A(p+1)m_{p,q} - Bpm_{p-1,q+1} - Cpm_{p-1,q} \\ -Aqm_{p+1,q-1} - Bp(q+1)m_{p,q} - Cqm_{p-1,q} \\ A(p+q+3)m_{p+1,q} + B(p+q+3)m_{p,q+1} + C(p+q+2)m_{p,q} \\ qm_{p,q-1} + (p+q+3)m_{p,q+1} \\ -pm_{p-1,q} + (p+q+3)m_{p+1,q} \\ pm_{p-1,q+1} - qm_{p+1,q-1} \end{pmatrix} \tag{1.57}$$

où A, B et C correspondent aux paramètres 3D du plan de la scène. Le théorème de Green permet de ne pas avoir à utiliser les gradients image qui pourraient induire des imprécisions dans la matrice d'interaction.

Le jeu de caractéristiques visuelles utilisé pendant l'asservissement est le suivant :

$$\mathbf{s} = (x_n, y_n, a_n, s_x, s_y, \alpha) \tag{1.58}$$

où $a_n = Z^* \sqrt{\frac{a^*}{a}}$ avec Z^* la distance entre la scène et la caméra et $a = m_{0,0}$ l'aire au sens photométrique de l'image. $x_n = \frac{m_{1,0}}{m_{0,0}} a_n$ et $y_n = \frac{m_{0,1}}{m_{0,0}} a_n$ correspondent aux centres de gravité de l'image selon les axes $\vec{\mathbf{x}}$ et $\vec{\mathbf{y}}$. Ces trois caractéristiques contrôlent les translations de la caméra. s_x, s_y et α contrôlent les rotations de la caméra. Une loi de contrôle classique peut alors être utilisée pour commander les déplacements de la caméra :

$$\mathbf{v} = -\lambda {\mathbf{L}_{\mathbf{s}^*}^{||}}^{-1} (\mathbf{s}^* - \mathbf{s}) \tag{1.59}$$

où ${\mathbf{L}_{\mathbf{s}^*}}^{||}$ est la matrice d'interaction calculée à la pose désirée uniquement. La modélisation de la matrice ${\mathbf{L}_{\mathbf{s}^*}^{||}}^{-1}$ permet de découpler le contrôle de la caméra.

L'utilisation du théorème de Green pour ne pas avoir à utiliser les gradients image implique une évaluation des pixels aux bords de l'image. Dans un premier temps [Bakthavatchalam 2013], cette évaluation était volontairement ignorée mais cela posait problème pendant l'asservissement lorsque des informations visuelles entraient ou sortaient du champ de vue de la caméra. Une amélioration de l'asservissement visuel basé moments photométriques a été proposée pour répondre à ce problème [Bakthavatchalam 2015]. L'idée de cette amélioration est

de venir pondérer les moments des images en donnant moins d'importance aux zones dans lesquelles les informations visuelles entrent et sortent (les pixels aux bords des images). La fonction de pondération a été soigneusement choisie de telle sorte qu'elle puisse être ajoutée à la formulation des moments photométriques et être en mesure de calculer une forme analytique de la matrice d'interaction.

1.3 Asservissement visuel virtuel

Le but des asservissements visuels virtuels [Sundareswaran 1998, Marchand 2002b] est le même que pour les asservissements visuels présentés précédemment, à savoir, déplacer une caméra vers une pose désirée en utilisant les informations visuelles retournées par celle-ci. Cependant, en asservissement visuel virtuel cette caméra n'est pas réelle mais virtualisée. Les images acquises par cette caméra ne sont donc pas des images numériques de la scène réelle mais elles sont le fruit de la projection d'un modèle virtuel 3D représentant la scène réelle. L'image désirée, quant à elle, est une image numérique (une image réelle par abus de langage) de la scène. Le but est de minimiser l'erreur entre l'image numérique acquise par la caméra réelle et la projection du modèle vue par la caméra virtuelle. À convergence, la pose de la caméra virtuelle est confondue avec la pose de la caméra réelle. L'asservissement visuel virtuel peut donc être considéré comme une méthode non-linéaire d'estimation de pose de caméra. Les applications de suivi d'objet et de réalité augmentée sont alors directement envisageables. Les différents asservissements visuels virtuels peuvent être répertoriés en fonction des primitives visuelles utilisées. Le choix de ces primitives dépend, a fortiori, directement du type de modèle 3D utilisé pour représenter la scène réelle.

1.3.1 Asservissement visuel virtuel basé primitives géométriques

Comme pour l'asservissement visuel basé primitives géométriques, le but ici est de minimiser l'écart entre les primitives extraites de l'image numérique désirée et les primitives résultantes de la projection d'un modèle virtuel représentant l'environnement dans lequel évolue la caméra réelle. Tout type de caractéristiques géométriques (points, lignes, cercles, cylindres...) peut être considéré dans la loi de commande dès lors qu'il est possible de calculer la matrice d'interaction correspondant à ce type de primitive. Il est possible de combiner plusieurs caractéristiques géométriques dans un même asservissement en empilant les primitives $\mathbf{s}$ dans un seul vecteur et en empilant respectivement les matrices d'interactions

$\mathbf{L}$ dans une grande matrice :

$$\begin{pmatrix} \dot{\mathbf{s}}_1 \\ \dot{\mathbf{s}}_2 \\ \dots \\ \dot{\mathbf{s}}_n \end{pmatrix} = \begin{pmatrix} \mathbf{L_1} \\ \mathbf{L_2} \\ \dots \\ \mathbf{L_n} \end{pmatrix} \mathbf{v} \tag{1.60}$$

Le type de modèle le plus utilisé pour représenter une scène en asservissements visuels virtuels est le modèle filaire [Marchand 2002a, Comport 2006]. C'est un modèle composé de sommets et d'arêtes représentant des caractéristiques visuelles à la géométrie rectiligne de la scène. Le principe général est de rechercher, dans la direction normale à une arête projetée du modèle, le contour réel le plus proche dans l'image numérique. Par conséquent, la primitive utilisée est la distance entre les points $\mathbf{s}(\mathbf{r})$ constituant la projection des arêtes du modèle filaire et les points constituant les contours détectés dans l'image désirée $\mathbf{s}^*$. Le bon fonctionnement de l'approche dépend alors grandement de la détection des contours de la scène dans l'image numérique désirée. Dans les applications de suivi ou de réalité augmentée, la détection de ces contours peut s'avérer difficile en raison d'occultations partielles de la scène ou de variations dans les images désirées acquises au cours du temps rendant les contours moins précis. C'est pourquoi, comme pour l'asservissement visuel photométrique (Section 1.2.2.1), la méthode peut être rendue plus robuste à l'aide d'un M-Estimateur [Comport 2003a, Dionnet 2007]. L'erreur $\mathbf{e}$ à réguler s'écrit alors :

$$\mathbf{e} = \mathbf{W}(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*) \tag{1.61}$$

avec $\mathbf{W}$, la matrice de pondération diagonale suivante :

$$\mathbf{W} = \begin{pmatrix} w_1 & & 0 \\ & \ddots & \\ 0 & & w_n \end{pmatrix} \tag{1.62}$$

où n est le nombre de primitives. Les poids w_i sont calculés à chaque itération et représentent le niveau de confiance des primitives qu'ils pondèrent. La valeur des poids est calculée statistiquement en fonction de la répartition des valeurs des erreurs de $\mathbf{e}$.

La caméra virtuelle est contrôlée à partir d'une loi de contrôle conçue pour tenter d'assurer une décroissance exponentielle découplée de l'erreur $\mathbf{e}$ autour de la position désirée :

$$\mathbf{v} = -\lambda(\mathbf{W}\mathbf{L_s})^{+}\mathbf{W}(\mathbf{s}(\mathbf{r}) - \mathbf{s}^*) \tag{1.63}$$

avec $\mathbf{L_s}$ la matrice d'interaction reliant les déplacements de la caméra par rapport à l'évolution des distances entre les points $\mathbf{s}(\mathbf{r})$ constituant la projection des arêtes

du modèle filaire et les points $\mathbf{s}^*$ constituant les contours détectés dans l'image désirée.

L'approche a été adaptée à la vision omnidirectionnelle [Marchand 2007, Caron 2012]. Pour cela, la distance point-droite a été reformulée dans le plan de l'image omnidirectionnelle et la matrice d'interaction associé à cette primitive a été établie pour ce modèle de caméra.

L'asservissement visuel virtuel peut également être utilisé dans une configuration "eye-to-hand". Par exemple, en ayant un modèle 3D représentant un robot humanoïde et un bras manipulateur, [Gratal 2013] parviennent à estimer l'allure de ces robots en minimisant la distance entre les contours extérieurs de la projection des modèles 3D et les contours réels des robots extraits des images numériques acquises par une caméra filmant les déplacements des robots. Une approche similaire est utilisée pour estimée et suivre la pose de l'organe effecteur d'un bras robotique [Gratal 2011].

1.3.2 Asservissement visuel virtuel dense

La représentation virtuelle d'un environnement peut être plus détaillée qu'un jeu de primitives géométriques. La géométrie de l'environnement ainsi que son aspect visuel peuvent être modélisés plus précisément. Ces types de modèles 3D peuvent alors être utilisés pour calculer la pose d'une caméra réelle en utilisant comme primitive visuelle l'apparence de l'environnement tel qu'il est perçu dans les images numériques réelles et dans les images virtuelles.

1.3.2.1 Modèle polygonal 3D avec texture photométrique

La géométrie d'une scène ou d'un objet peut être modélisée plus précisément en représentant sa spatialité sous la forme d'un ensemble de polygones. Une texture ou une couleur est généralement appliquée sur chacun des polygones, donnant ainsi au modèle virtuel un aspect visuel plus proche de la réalité d'un point de vue photométrique.

Ce type de modèle a été utilisé, par exemple, dans [Caron 2014] où une partie des rues du 12^e arrondissement de Paris a été modélisée (Figure 1.10 (a)). Les bâtiments sont modélisés par de simples parallélépipèdes rectangles sur lesquels sont plaquées des images issues de prises de vues aériennes.

Le critère d'optimisation choisi ici est l'information mutuelle IM (eq. 1.41). À la différence de l'asservissement visuel basé information mutuelle (Section 1.2.2.2), la maximisation de l'IM ne se fait pas sur deux images numériques provenant d'un même capteur mais entre une image numérique réelle $\mathbf{I}$ (Figure 1.10 (b)) et les images virtuelle $\mathbf{I_v}(\mathbf{r})$ (Figure 1.10 (c)). Une image virtuelle $\mathbf{I_v}(\mathbf{r})$ est le fruit de la projection du modèle 3D à partir de la pose de caméra $\mathbf{r} = (t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)$.

(a) (b) (c)

FIGURE 1.10: Modèle polygonale 3D avec texture photométrique des rues du 12^e arrondissement de Paris utilisé dans [Caron 2014] (a), une image numérique réelle prise dans l'une des rues (b), et une image virtuelle générée dans le modèle à la même pose de caméra (c)

L'image $\mathbf{I_v}(\mathbf{r})$ est générée par un moteur de rendu 3D. L'une des étapes composant la génération d'un rendu est le Z-Buffer. Le Z-Buffer permet de déterminer quels éléments de la scène sont ou ne sont pas visibles, lesquels sont cachés par d'autres et dans quel ordre l'affichage des polygones du modèle doit s'effectuer. Il est possible d'extraire de cette étape une carte des profondeurs de la scène vue à partir de la pose de la caméra virtuelle. Contrairement à un asservissement visuel où les profondeurs ne sont pas connues, ici, les profondeurs provenant du moteur de rendu sont utilisées pour calculer la matrice d'interaction $\mathbf{L_x}$ reliant les déplacements des points dans l'image par rapport aux mouvements de la caméra.

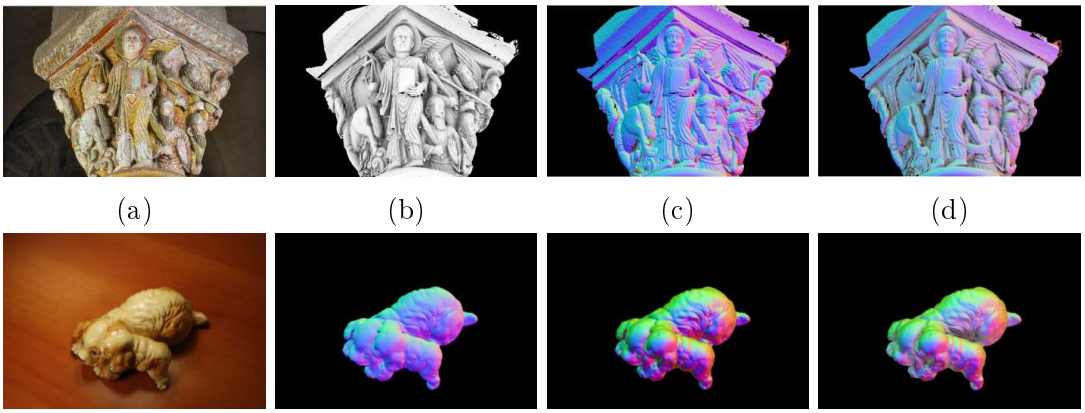
1.3.2.2 Modèle polygonal 3D avec texture géométrique

Des méthodes, souvent chronophages, permettent de créer un modèle 3D d'une scène ou d'un objet dont la géométrie est très fidèle à la réalité. La géométrie et l'aspect visuel de la scène ne sont généralement pas acquis simultanément mais sont relevés par deux procédés distincts.

Plutôt que d'utiliser l'information photométrique du modèle, il est possible d'utiliser les informations géométriques de celui-ci pour générer différents rendus de la scène modélisée. Ces types de rendus sont utilisés par [Corsini 2009] dans le but d'estimer la pose d'une caméra réelle à l'origine d'une image numérique. L'estimation de cette pose est réalisée dans le cadre d'un asservissement visuel virtuel dans lequel le critère d'optimisation est l'information mutuelle IM (eq. 1.41). L'image désirée reste une image numérique acquise par une caméra réelle. Cependant les images virtuelles sont le fruit de la projection du modèle coloré en fonction de sa géométrie. Différentes caractéristiques liées à la géométrie du modèle 3D sont comparées dans [Corsini 2009] :

— Normales des surfaces : la droite normale à une surface en un point est la droite orthogonale au plan tangent en ce point (Figures 1.11c et 1.11f).

— Réflexion des surfaces : la réflexion d'une surface en un point dépend de la normale en ce point et de la position de la caméra virtuelle (Figure 1.11g).
— Occultations ambiantes : cette méthode de rendu permet d'assombrir les zones naturellement difficiles d'accès à la lumière (Figure 1.11b). Cela a pour effet de faire apparaître le relief des objets, là où les objets sans l'application de cette technique (ou une autre technique plus élaborée), apparaîtraient entièrement plats.



(a) (b) (c) (d)

(e) (f) (g) (h)

FIGURE 1.11: Différents modèles de représentation 3D d'objets : Image numérique d'un chapiteau (a), les occultations ambiantes (b), les normales (c) et une combinaison des occultations ambiantes et des normales calculées sur modèle 3D du chapiteau. Image numérique d'un chien en porcelaine (e), les réflexions (f), les normales (g) et une combinaison des occultations ambiantes et des réflexions calculées sur modèle 3D du chien en porcelaine

Différentes expérimentations ont permis de mettre en évidence qu'en fonction de la scène, il est parfois préférable d'utiliser l'une ou l'autre des caractéristiques géométriques du modèle ou encore une combinaison de deux caractéristiques géométriques (Figures 1.11d et 1.11h).

1.4 Conclusion

L'asservissement visuel est un domaine de recherche qui ne cesse d'évoluer. De nombreuses méthodes et améliorations ont été proposées et un grand nombre de primitives visuelles sont désormais utilisables pour déterminer la loi de commande. Ces informations visuelles peuvent être globalement classées en deux catégories : les caractéristiques géométriques et photométriques. Contrairement aux caractéristiques géométriques, le photométrique a l'avantage de ne nécessiter aucun traitement d'image de type détection, appariement, suivi ou encore

segmentation. De plus, grâce à la redondance d'informations visuelles, les asservissements photométriques possèdent une excellente précision à convergence. Tout d'abord, il a été proposé d'utiliser directement les intensités de tous les pixels des images comme caractéristiques photométriques. Cette primitive s'est montrée robuste aux approximations sur les profondeurs de la scène, aux occultations locales et peut être utilisée dans des environnements spéculaires et peu texturés. La loi de commande étant directement liée aux intensités des images, la trajectoire empruntée par la caméra au cours de l'asservissement peut s'avérer être assez chaotique. Par la suite, différentes améliorations ont été proposées comme, par exemple, d'intégrer un M-Estimateur afin de rendre l'asservissement plus robuste aux occultations. Le principe consiste à pondérer les valeurs d'erreur entre les pixels de l'image courante et de l'image désirée en se basant sur l'écart type de ces erreurs. Pour pallier les problèmes liés aux variations d'illumination, il a été proposé d'adapter les intensités de l'image désirée en fonction de celles de l'image courante tout au long de l'asservissement. D'autres méthodes proposent également d'utiliser l'intégralité des pixels mais de manière indirecte. Par exemple, en cherchant à maximiser l'information mutuelle calculée entre l'image désirée et l'image courante. Cette mesure de similarité s'est montrée robuste aux bruits, aux réflexions spéculaires et s'avère être intéressante pour comparer deux images ayant des modalités différentes. Cependant, toutes ces méthodes possèdent un domaine de convergence relativement étroit. En effet, la pose optimale de la caméra étant considérée comme la solution d'un problème d'optimisation non-linéaire, converger correctement vers cette solution dépend directement de la distance entre la pose initiale de la caméra et la pose désirée. Plus concrètement, l'image obtenue à la pose désirée et l'image obtenue à la pose initiale doivent avoir un recouvrement d'informations photométriques suffisamment important pour que la caméra puisse converger vers la pose souhaitée. Récemment, les moments photométriques et les noyaux ont permis d'élargir le domaine de convergence et ont l'avantage de pouvoir définir des lois de commande découplées rendant les trajectoires empruntées par la caméra plus directes.

Parallèlement à ces avancées, de nouvelles technologies de numérisation 3D ont vu le jour. Il est désormais possible de mesurer rapidement et avec une grande précision la structure spatiale d'un environnement. Ces appareils de mesure sont à l'origine de nouvelles données telles que les nuages de points 3D colorés de grande taille. Il devient alors intéressant d'utiliser ce type de représentation de l'environnement dans le cadre d'asservissements visuels virtuels afin d'étudier l'apport de ces modèles 3D ainsi que les limites des méthodes de l'état de l'art sur ces données. Des améliorations et des contributions dans ce domaine sont alors envisageables.

Chapitre 2

Asservissements visuels virtuels basés nuages de points colorés

Sommaire

Ce chapitre a pour but d'étendre l'asservissement visuel photométrique (Section 1.2.2.1) à l'asservissement visuel virtuel photométrique. Nous avons vu qu'une scène peut être virtuellement représentée de différentes manières (modèle filaire, modèle 3D texturé...) et que cette représentation a une influence directe sur l'asservissement. Ces dernières années, les technologies de numérisation 3D sont devenues très efficaces et permettent aujourd'hui de mesurer rapidement et avec précision la structure spatiale d'une scène réelle. C'est pourquoi nous proposons d'utiliser les modèles 3D obtenus grâce à ces nouveaux outils comme représentation de l'environnement dans le cadre d'asservissements visuels virtuels.

Le projet E-Cathédrale se focalise sur la numérisation de la cathédrale d'Amiens, dans le but d'en obtenir un modèle complet et très précis. Nos travaux exploitant les relevés obtenus durant les campagnes d'acquisition du projet E-Cathédrale, nous présentons plus en détails ce projet ainsi que la lasergrammétrie, sa principale méthode de modélisation de l'environnement.

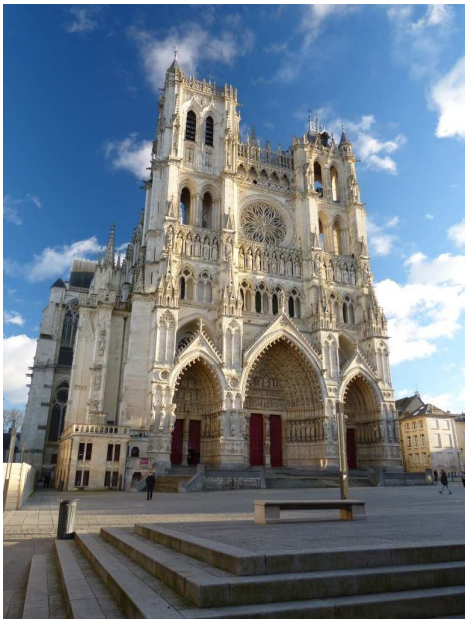


FIGURE 2.1: La façade occidentale de la cathédrale Notre-Dame d'Amiens. Photographie prise à partir du parvis Ouest

2.1 Modélisation de l'environnement

2.1.1 Projet E-Cathédrale

Le patrimoine architectural est composé de vestiges considérables de notre passé. Cet héritage culturel est une fenêtre sur la vie de nos ancêtres mettant en lumière une période, une société ou encore un moment précis de notre histoire. Ce patrimoine constitue un lien historique, artistique et bien souvent spirituel d'une grande importance. Ces témoins du temps qui passe comprennent, entre autres, des monuments, des bâtiments et des lieux historiques. Ces sites peuvent régulièrement être soumis à des conditions environnementales difficiles, à des détériorations causées par l'homme et sont bien souvent victimes des années. Il est par conséquent dans notre intérêt et de notre devoir d'assurer leur protection, leur préservation et leur restauration. C'est pourquoi la sauvegarde du patrimoine est un axe de recherches important et en perpétuelle évolution.

La cathédrale Notre-Dame d'Amiens fait partie de ce patrimoine architectural à sauvegarder et à entretenir afin d'être exposé au plus grand nombre. Avec une hauteur sous voûte de 42,30 mètres pour un volume intérieur de l'ordre de 200 000 mètres cube, les dimensions hors-normes de la cathédrale d'Amiens (Figure 2.1) en font la cathédrale complète (à la différence de celle de Beauvais) la plus grande de France. Cet édifice au style typiquement gothique a vu ses travaux de construction débuter en 1220 et s'étendre sur plusieurs siècles. La

cathédrale est inscrite en France comme monument historique depuis 1862 et est entrée au patrimoine mondiale de l'UNESCO [1] en 1981. Malgré ces distinctions et son imposante architecture, avant le lancement du projet il n'existait aucun plan complet et précis de l'édifice. C'est de ce constat qu'est né le programme de recherche E-Cathédrale [2].

Le programme de recherche E-Cathédrale a vu le jour en 2010 pour une durée de 15 ans. L'objectif de ce programme est de travailler sur la réalisation et sur l'exploitation d'une maquette numérique de la cathédrale Notre-Dame d'Amiens.

Cette reconstitution virtuelle a plusieurs finalités : une vocation scientifique dans un premier temps, de par la modélisation elle-même de la maquette, la surveillance de l'évolution du monument, le calcul des structures ou encore dans l'analyse et dans l'étude artistique ou historique de l'édifice. Le programme a également un rôle sociétal important puisque la maquette pourra servir à archiver et sauvegarder le patrimoine immense scellé dans ce monument historique. Enfin, E-Cathédrale possède aussi une finalité culturelle puisque la maquette permettra de faciliter l'accès à la culture par le biais du numérique ou encore de sensibiliser le grand public via une approche ludique et attractive.

Pour répondre à ces attentes, la maquette virtuelle de la cathédrale d'Amiens se doit d'être complète, aussi bien de l'extérieur que de l'intérieur, et la plus fidèle en termes de géométrie et d'aspect. Tous les ans depuis le lancement du programme de recherche, durant une semaine, différentes méthodes sont employées afin de relever la structure du bâtiment. Des prises de vue aériennes et terrestres sont acquises afin de faire de la photogrammétrie et des campagnes de numérisation par lasergrammétrie sont effectuées. En effet, de récentes avancées technologiques dans ce domaine ont permis le développement de méthodes et d'outils, comme les scanners laser 3D, capables de mesurer avec précision la structure spatiale d'un environnement. Cette capacité de générer rapidement et précisément des nuages de points a fait des scanners laser 3D des outils de plus en plus utilisés en documentation historique. Étant données les dimensions imposantes de la cathédrale, la lasergrammétrie semblait une approche judicieuse afin de relever la géométrie de l'édifice. Dans les travaux présentés dans cette thèse, ce sont ces données relevées par les scanners laser qui sont utilisées. C'est pourquoi, la lasergrammétrie est présentée plus en détail dans ce qui suit.

2.1.2 Lasergrammétrie

De l'industrie à la topographie classique, en passant par l'informatique ou les sciences judiciaires, sans oublier le domaine du patrimoine : qu'elle soit fixe ou mobile, terrestre ou aérienne, la lasergrammétrie est employée dans de nombreux

1. Organisation des Nations Unies pour l'éducation, la science et la culture
2. www.mis.u-picardie.fr/E-Cathedrale/

domaines. La méthode consiste à effectuer automatiquement un balayage laser afin de relever la structure spatiale d'un objet ou d'une scène environnante. Les outils réalisant cette numérisation 3D sont généralement appelés scanners laser. Dans le cadre du programme de recherche E-Cathédrale, nous nous intéresserons plus particulièrement à la numérisation 3D à partir de balayages laser effectuée à partir de scanners laser terrestres fixes.

Les scanners laser émettent un rayon laser afin de mesurer directement, sans contact, la distance séparant l'appareil d'un objet en un point. Le laser est monté sur une tête rotative pouvant effectuer des rotations horizontales et pointe vers un miroir rotatif pouvant réaliser des rotations verticales. Cette configuration permet au scanner d'orienter très rapidement le rayon laser dans toutes les directions. L'émission du rayon laser s'effectue à une fréquence élevée, ce qui permet de réaliser un balayage rapide de la scène qui entoure le scanner et ainsi de relever jusqu'à un million de points par seconde. Les rotations nécessaires à l'orientation du rayon sont mesurées avec précision et permettent de calculer les coordonnées 3D des points d'impact du laser sur les surfaces de la scène.

Il existe plusieurs types de scanner laser, différentiables selon leur méthode de mesure de distance. Les trois principaux types de scanner sont les scanners à triangulation, à différence de phase et à temps de vol (ou à impulsion). Ces deux derniers types de scanner sont utilisés dans le programme de recherche E-Cathédrale.

— Scanner à temps de vol : Ce scanner détermine la distance entre sa position et une surface en mesurant le temps écoulé entre la transmission et la réception d'une impulsion laser orientée vers cette surface. Cette méthodologie implique au scanner de ne pas pouvoir émettre une nouvelle impulsion avant d'avoir reçu l'écho de la précédente. Par conséquent, sa capacité à mesurer de faibles intervalles de distances est directement liée à sa capacité à mesurer de petits intervalles de temps. Ce type de scanner est plus adapté aux relevés à moyennes et longues portées. Le scanner à temps de vol utilisé pendant les campagnes de numérisation de la cathédrale d'Amiens est un Leica C10 (Figure 2.2a).

— Scanner à différence de phase : Ce scanner émet un rayon laser dont le signal est modulé selon une sinusoïde. Pour mesurer la distance entre une surface et le scanner, ce dernier compare les phases entre les ondes émises et les ondes reçues. Le signal sinusoïdal est reçu avec un temps de retard proportionnel à la différence de phase dans laquelle intervient la fréquence de modulation connue du signal envoyé. Cette méthodologie nécessite un signal laser de forte intensité. Pour ces raisons, ce type de scanner est plus adapté aux relevés à courtes et moyennes portées (inférieur à 100 mètres). Le scanner à différence de phase utilisé pendant les campagnes de numérisation de la cathédrale d'Amiens est un FARO Focus 3D (Figure 2.2b).

(a) (b)

FIGURE 2.2: Scanners utilisés dans le cadre du programme de recherche E-Cathédrale : scanner à temps de vol Leica C10 (a) et scanner à différence de phase Faro Focus 3D (b)

Qu'il soit à différence de phase ou à temps de vol, l'ensemble des mesures d'une scène relevé par un scanner constitue un nuage de points 3D. Ce nuage de points représente les surfaces des objets présents autour du scanner pendant la numérisation. Ces points 3D sont définis dans un repère relatif à la position du scanner.

En plus des coordonnées cartésiennes des points, le scanner retourne également l'intensité de retour du signal laser de chaque relevé. Cette intensité dépend d'un certain nombre de paramètres : distance et orientation entre la surface de l'objet et le scanner, type de surface, la matière de l'objet ou encore sa couleur. Cette valeur possède différents noms dans la littérature, nous l'appellerons réflectance.

La plupart des scanners laser récents sont équipés d'une caméra numérique intégrée ou pouvant être ajoutée à l'appareil. Cette caméra est également montée sur la tête rotative du scanner de façon à pouvoir orienter son axe optique dans toutes les directions. Après avoir effectué l'acquisition du nuage de points 3D, le scanner utilise la caméra afin de prendre un nombre d'images suffisamment important pour couvrir l'ensemble de la scène numérisée et créer une image équirectangulaire (Section 1.1.1.3). À partir de cette image, une couleur RVB (rouge, verte, bleue) est associée à chaque point 3D du nuage.

Pour faciliter l'exploitation d'un nuage de points complet d'une scène, plu-

sieurs étapes sont à respecter pendant la numérisation.

L'acquisition de la structure spatiale d'une scène à partir d'une position est appelée une station. Dans la grande majorité des cas, une scène ne peut pas être numérisée à partir d'une seule station, et ce pour différentes raisons : le champ de vision du scanner peut être limité, la scène peut être de dimension importante ou d'architecture complexe, ou encore, des obstacles (fixes ou mobiles) peuvent empêcher la numérisation directe de certaines zones. Dans ces conditions, pour couvrir totalement un environnement, plusieurs points de vue différents sont nécessaires. Chaque nuage est alors exprimé dans un repère relatif à la position du scanner lors de son acquisition. La première étape est donc d'assembler l'intégralité des nuages de points dans un repère commun. Cet assemblage (ou ce recalage) se fait par la détection de points de correspondance entre les différents nuages. Ces points peuvent être des amers naturels caractéristiques de la scène numérisée ou encore des cibles placées dans la scène pendant les acquisitions. L'utilisation de cibles pendant la numérisation rend les acquisitions plus difficiles car trois cibles fixes (deux si l'angle du scanner avec le sol ne change pas entre deux stations) doivent être visibles par le scanner entre deux stations successives. Cependant cette vigilance pendant les relevés facilite grandement l'étape d'assemblage des nuages de points. Bien souvent, l'assemblage à partir de cibles ou d'amers naturels est affiné à l'aide d'un second recalage automatique des nuages basé par exemple sur l'Iterative Closest Point [Besl 1992].

Pour la numérisation de la cathédrale d'Amiens, les dimensions importantes et l'architecture complexe de l'édifice nous obligent à réaliser un grand nombre de stations afin de couvrir le bâtiment dans sa totalité. Par exemple, en se focalisant sur le portail sud de la cathédrale, la figure 2.3a montre quatre nuages de points résultant de quatre stations différentes, la figure 2.3b, quant à elle, montre l'assemblage de ces quatre nuages.

L'assemblage mène à un très grand nombre de mesures 3D très précises couvrant le portail dans son intégralité mais la qualité des caméras utilisées sur les scanners laser mène à un mauvais aspect visuel dû à plusieurs problèmes identifiés.

Premièrement, la définition de la caméra est bien moindre que celle du scanner. Par conséquent, plusieurs points 3D seront colorés à partir du même pixel provenant de l'image équirectangulaire. Ce phénomène crée un effet de flou sur le nuage de points, qui est alors loin d'être représentatif de l'aspect réel de la scène. Deuxièmement, la dynamique photométrique de la caméra est plutôt basse, menant à des problèmes d'exposition classiques. Pour terminer, puisqu'en pratique plusieurs acquisitions à partir de plusieurs positions doivent être réalisées pour couvrir un édifice de grande taille, la combinaison de ces acquisitions mène presque toujours à des couleurs incohérentes.

Utiliser une caméra numérique de très grande qualité à la place des caméras

(a) (b)

FIGURE 2.3: Acquisitions et assemblage : quatre nuages de points bruts obtenus à partir des différentes positions du scanner à laser (a) et le nuage de points issu de leur recalage (b)

existantes sur les scanners améliorerait l'aspect visuel mais seulement pour une profondeur limitée de perception, et au prix d'un dispositif bien plus gros et d'un temps d'acquisition plus long.

2.2 Pré-traitements des nuages de points

Lorsque l'environnement à scanner est de grande envergure (villes, monuments historiques, bâtiments industriels...), le nombre de point 3D peut rapidement devenir immense. Par conséquent, générer une image virtuelle à partir de ces points devient une opération extrêmement chronophage. De plus, à cause des problèmes susmentionnés liés aux acquisitions, les images obtenues ont une mauvaise qualité photométrique, et ne représentent donc pas parfaitement l'environnement réel (Figure 2.3b). Pour ces raisons, nous proposons deux étapes de pré-traitements des nuages de points afin de rendre le modèle exploitable.

2.2.1 Homogénéisation des couleurs du modèle

Notons $P_{i,j}(\mathbf{X}, \mathbf{C})$ un point 3D coloré j d'un nuage de points i, composé de ses coordonnées $\mathbf{X} = (X, Y, Z)^T$ et de sa couleur $\mathbf{C} = (R, V, B)^T$. Un nuage de points acquis par la station i est exprimé par $\mathbf{PC}_i = \{P_{i,j}(\mathbf{X}, \mathbf{C}) / \forall j \in [\![0, N_i - 1]\!]\}$, N_i étant le nombre de points acquis par la station i. Nous notons $\mathbf{PCm} =$

$\{\mathbf{PC}_i/\forall i \in [\![0, N_j - 1]\!]\}$ l'assemblage de N_j nuages de points obtenus par des stations distinctes à différents moments de la journée.

Pour pallier les problèmes rendant les images virtuelles photométriquement mauvaises, nous nous sommes inspirés de [Tian 2002] afin d'homogénéiser la couleur des nuages de points PC_i entre eux. Dans ces travaux, la correction des couleurs est effectuée sur des images numériques acquises afin de créer des panoramas par mosaïquage. Les images numériques utilisées peuvent avoir été prises sous différents points de vue et à différents moments de la journée et donc sous différents éclairages. Les différences d'illumination entre les images numériques acquises par l'appareil photo numérique (APN) sont considérées comme suivant une transformation linéaire $\mathbf{T}_{(3\times3)}$ des couleurs. La comparaison de différentes formes de $\mathbf{T}_{(3\times3)}$ par [Tian 2002] a permis de mettre en évidence les meilleures performances du modèle linéaire compensé par une transformation affine. Cette matrice de transformation est calculée par :

$$\mathbf{T}_{(3\times3)} = [\mathbf{J_1}^{\mathrm{T}}\mathbf{J_1}]^{-1}\mathbf{J_1}^{\mathrm{T}}\mathbf{J_2} \tag{2.1}$$

$\mathbf{J_1}_{(3\times N)}$ et $\mathbf{J_2}_{(3\times N)}$ contiennent les N valeurs RVB des pixels d'une partie de la scène que les deux images ont en commun, une zone de chevauchement [Tian 2002]. Ce chevauchement permet de considérer que les pixels de $\mathbf{J_1}$ et $\mathbf{J_2}$ sont en corrélation, autrement dit, les pixels aux coordonnées $\mathbf{u} = (u, v) \in \mathbf{J_1} \wedge \mathbf{J_2}$ des deux images sont censés avoir la même valeur RVB. $\mathbf{J_1}$ et $\mathbf{J_2}$ sont donc de taille $(3 \times N)$, N étant le nombre de pixels communs d'une zone des images. La matrice $\mathbf{T}$ ainsi obtenue est de taille (3×3), chaque colonne permettant de corriger les trois canaux RVB de l'image $\mathbf{J_1}$. S'ajoute à cela la compensation affine calculée par :

$$\begin{pmatrix} R_{affine} \\ V_{affine} \\ B_{affine} \end{pmatrix} = \begin{pmatrix} moy(R2) \\ moy(V2) \\ moy(B2) \end{pmatrix} - \mathbf{T}_{(3\times3)} \times \begin{pmatrix} moy(R1) \\ moy(V1) \\ moy(B1) \end{pmatrix} \tag{2.2}$$

où $moy(R1)$ et $moy(R2)$ correspondent respectivement à la moyenne des valeurs du canal rouge de l'image 1 et de l'image 2. Il en est de même pour les canaux vert et bleu.

Nous avons étendu cette approche pour corriger les couleurs $\mathbf{C}_{(i,j)}$ des points 3D composant nos $\mathbf{PC}_i$. $\mathbf{J_1}$ et $\mathbf{J_2}$ ne contiennent plus les valeurs RVB de portions d'images mais les couleurs $\mathbf{C}_{(i,j)}$ de points qui se chevauchent appartenant à deux nuages $\mathbf{PC}_i$. Les équations présentées précédemment restent inchangées. Nous choisissons visuellement parmi les $\mathbf{PC}_i$ un nuage de points de référence que l'on appelle $\mathbf{PC}_{ref}$ et les couleurs des autres nuages sont, l'une après l'autre, automatiquement uniformisées. Comme pour les images numériques de [Tian 2002], la matrice de transformation linéaire $\mathbf{T}$ est calculée à partir des couleurs des points

qui se chevauchent entre $\mathbf{PC}_{ref}$ et le nuage ciblé. Pour remplir $\mathbf{J_1}$ et $\mathbf{J_2}$, nous cherchons pour chaque point de $\mathbf{PC}_{ref}$ son point voisin le plus proche appartenant au nuage ciblé dans une sphère de faible rayon. S'il existe un point dans cette sphère, alors les deux points sont considérés comme confondus, le point du nuage de référence est ajouté à $\mathbf{J_1}$ et le point du nuage cible est ajouté à $\mathbf{J_2}$.

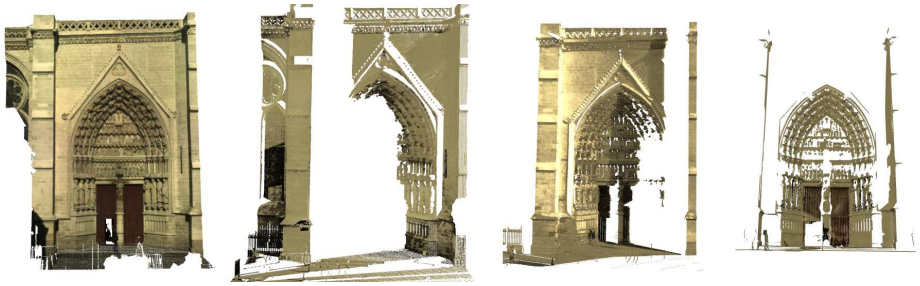
A l'exception du nuage $\mathbf{PC}_i$ choisi pour être le nuage de référence $\mathbf{PC}_{ref}$, les couleurs des $N_i - 1$ nuages de points composant **PCm** sont uniformisées avec la méthode décrite ci-dessus. Les couleurs $\mathbf{C}u$ des points des nuages uniformisées $\mathbf{PCu}_i$ sont donc calculées par :

$$\mathbf{C}u_{i,j} = \mathbf{C}_{i,j} \times T + RVB_{affine} \tag{2.3}$$

Les $N_i - 1$ nuages de points aux couleurs uniformisées $\mathbf{PCu}_i$ et le nuage $\mathbf{PC}_{ref}$ sont fusionnés pour former le nuage **PCum**. Les images virtuelles générées à partir du modèle aux couleurs uniformisées sont donc moins perturbées et sont plus cohérentes avec la scène réelle.

La figure 2.4 montre une comparaison avant/après l'homogénéisation des couleurs de quatre nuages de points du portail sud de la cathédrale.

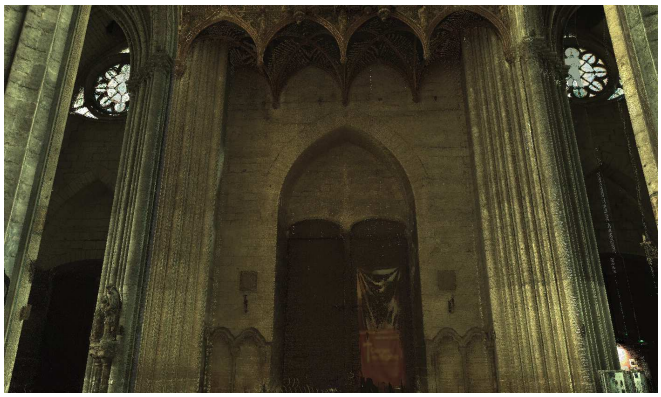
(a)



(b)

FIGURE 2.4: Homogénéisation des couleurs : quatre nuages de points bruts obtenus à partir des différentes positions du scanner à laser autour du portail sud de la cathédrale (le premier nuage correspond au nuage choisi comme référence) (a) et les mêmes nuages de points après l'homogénéisation de leurs couleurs (b)

(a)



(b)

FIGURE 2.5: Homogénéisation des couleurs : assemblage des nuages de points de la porte intérieur Est avant (a) et après (b) l'homogénéisation des couleurs

Les différences importantes de teinte entre les nuages avant l'homogénéisation des couleurs (Figure 2.4a) est principalement dues aux heures auxquelles les stations ont été effectuées et par conséquent à l'ensoleillement du portail. Même si le fait d'être en intérieur amenuise ces effets indésirables, la figure 2.5 montre que ces différences d'illuminations sont tout aussi importantes à corriger lorsque les nuages de points ont été acquis à l'intérieur de la cathédrale.

2.2.2 Structuration spatiale du modèle

Pour faire face à la grande quantité de données retournée par les scanners laser, nous proposons un arrangement des nuages de points pour ne considérer que les points 3D utiles du modèle selon le point de vue de la caméra dans l'environnement virtuel. Nous ne souhaitons pas utiliser l'intégralité des points 3D du modèle pour générer une image virtuelle mais uniquement utiliser les points considérés comme pouvant être visibles à partir de la position de la caméra

virtuelle.

Un nuage de points organisé **OPC** est le nom donné à un nuage de points qui a la même structure qu'une image, où les données sont réparties en lignes et en colonnes. Un **OPC** peut être considéré comme deux matrices, la première qui est une image d'intensité et la seconde qui contient les coordonnées des points 3D visibles dans cette image. Nous exprimons la génération d'un nuage de points organisé d'une scène par $\mathbf{OPC}(pr_x(\mathbf{PCum}),{}^{c}\mathbf{M}_s)$ où pr_x représente le type de projection de la caméra virtuelle, ${}^{c}\mathbf{M}_s$ est la pose de la caméra dans la scène 3D représentée par **PCum**, l'assemblage des nuages de points aux couleurs uniformisées dans lequel évolue la caméra virtuelle. Nous proposons de créer une base de données de nuages de points organisés, puis en fonction de la position de la caméra virtuelle de n'utiliser qu'un seul de ces **OPC** au lieu du modèle complet **PCum**. Pour créer la base de données d'**OPC**, l'espace dans lequel nous souhaitons pouvoir nous déplacer virtuellement dans le modèle doit être déterminé (par exemple, l'intérieur du modèle de la cathédrale). Cet espace est régulièrement partitionné selon une grille 3D. Une caméra équirectangulaire est placée au centre des k cases de cette grille, pour chaque ${}^{c_k}\mathbf{M}_s$ un nuage de points organisé est généré : $\mathbf{OPC}_k(pr_e(\mathbf{PCum}),{}^{c_k}\mathbf{M}_s)$ (Figure 2.6). Les points 3D contenus dans

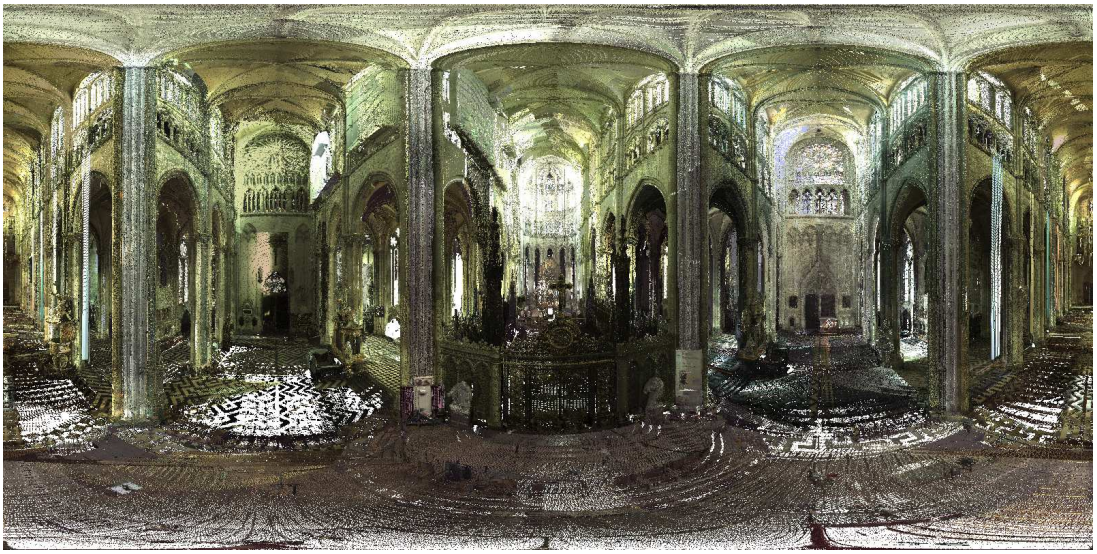


FIGURE 2.6: Exemple de nuage de points organisé généré à l'aide d'une caméra virtuelle équirectangulaire dans le modèle de la cathédrale d'Amiens

l'$\mathbf{OPC}_k$ sont les seuls points visibles depuis la position ${}^{c_k}\mathbf{M}_s$. Tous les nuages de points organisés ainsi que les positions à partir desquelles ils ont été générés sont sauvegardés dans une base de données. Dès lors, au lieu d'utiliser le modèle complet $\mathbf{PC_m}$, la base de données complète est chargée et les ${}^{c_k}\mathbf{M}_s$ sont placées dans un kd-tree. Pour générer une image virtuelle à partir d'une pose de caméra

${}^{c}\mathbf{M}_s$, une rapide recherche dans le kd-tree permet de trouver l'$\mathbf{OPC}_k$ créé depuis la position ${}^{c_k}\mathbf{M}_s$ la plus proche de ${}^{c}\mathbf{M}_s$. Enfin, l'image virtuelle n'est créée qu'en projetant les points 3D appartenant à cette $\mathbf{OPC}_k$. Cette structuration spatiale du modèle permet de générer des images virtuelles très rapidement et contenant quasiment la même information visuelle qu'en utilisant le modèle complet. Un nuage de points organisé étant créé à partir d'une pose de caméra fixe ${}^{c_k}\mathbf{M}_s$, lorsque cet $\mathbf{OPC}_k$ est utilisé pour générer les images virtuelles, plus la pose de la caméra s'éloigne de la pose ${}^{c_k}\mathbf{M}_s$, plus il y a de zones vides dans les images virtuelles. Pour réduire ces zones, il est possible de réduire le pas de la grille 3D. En pratique, la quantité d'informations visuelles perdues est minime comparé à la redondance d'information photométrique apportée par le reste de l'image, ces zones vides ne sont donc pas un problème compte tenu de l'utilisation que nous faisons de ces images.

Dans la suite, pour simplifier et alléger les écritures, même si ce n'est pas mentionné, une image virtuelle **Iv** est toujours générée à partir d'un seul **OPC** de la base de données de nuages de points organisés dont les couleurs du modèle complet ont été préalablement homogénéisées.

2.3 Calcul de pose basé points

L'estimation de la pose d'une caméra réelle à l'origine d'une image numérique en utilisant un ensemble de points se ramène à un problème d'optimisation non-linéaire que l'on propose de résoudre en utilisant le même formalisme : l'asservissement visuel virtuel.

Dans notre cas, nous possédons une représentation virtuelle de l'environnement dans lequel l'image numérique a été prise, le modèle **PCum** composé de nuages de points colorés. À partir d'une pose initiale, l'asservissement visuel virtuel fait évoluer la pose d'une caméra virtuelle par rapport à cet environnement. L'évolution de la pose dépend des déplacements des primitives, ici des points, dans l'image virtuelle.

Il est donc nécessaire d'avoir un jeu d'appariement de points 2D entre l'image numérique **I** et l'image virtuelle $\mathbf{Iv}(\mathbf{r})$ où $\mathbf{r}$ est la pose initiale de la caméra virtuelle. La détection et la mise en correspondance de ces points peuvent être effectuée manuellement ou automatiquement. L'appariement automatique de primitives visuelles ne pourrait être réalisé sans l'homogénéisation des couleurs des nuages de points composant **PCum** présentée précédemment. En effet, les différences de couleurs entre les nuages de points provenant de différentes stations créent des perturbations sur les images virtuelles générées (Figure 2.3). Ces dégradations jouent un rôle important sur les descripteurs des points d'intérêt détectés et rendent leur mise en correspondance impossible.

Soit $\mathbf{I_v}(\mathbf{r})$ l'image virtuelle obtenue à partir de la projection de la caméra virtuelle. Nous sommes en mesure de remonter aux coordonnées 3D de ces points. Nous nous retrouvons alors avec un problème d'asservissement visuel virtuel basé points classique, la régulation de l'erreur prend alors la forme :

$$\mathbf{e} = \mathbf{x}(\mathbf{r}) - \mathbf{x}^* \tag{2.4}$$

où $\mathbf{x}(\mathbf{r})$ est un vecteur contenant la position de la projection des points 3D détectés dans l'image virtuelle initiale à la pose de la caméra $\mathbf{r}$ et $\mathbf{x}^*$ correspond à la position des points 2D appariés dans $\mathbf{I}$.

L'erreur ne varie qu'en fonction de $\mathbf{x}$ et mène à la matrice d'interaction suivante :

$$\dot{\mathbf{x}} = \mathbf{L_x v} \tag{2.5}$$

où $\mathbf{X}$ est le point 3D dans le repère caméra obtenu à partir des points 2D de l'image virtuelle initiale et de la projection inverse de la caméra.

La matrice d'interaction calculée, une méthode d'optimisation telle qu'une descente de gradient, un Gauss-Newton ou un Levenberg-Marquardt est utilisée pour calculer la loi de commande de la caméra virtuelle permettant de réguler l'erreur $\mathbf{e}$ à zéro. À la fin de l'optimisation, la projection des points 3D $\mathbf{X}$ du modèle converge vers la position des points 2D de l'image numérique. Par conséquent, la pose de caméra virtuelle ainsi obtenue est plus proche de la pose de la caméra à l'origine de l'image numérique réelle.

2.4 Calcul de pose dense

L'estimation de pose de caméra peut également être exprimée comme un problème d'optimisation dans le cadre d'un asservissement visuel virtuel mais, cette fois, en utilisant toute l'information contenue dans les images. Cependant, contrairement à l'asservissement visuel photométrique [Collewet 2008], nous ne cherchons pas à minimiser la différence entre deux images réelles provenant d'un même capteur de vision. Ici, nous cherchons à minimiser la différence entre une image numérique $\mathbf{I}$ et l'image virtuelle $\mathbf{Iv}$ obtenue à partir d'une caméra virtuelle évoluant dans une représentation 3D de l'environnement réel. Dans notre cas, le modèle 3D représentant l'environnement réel est le fruit d'acquisitions de mesures à l'aide de scanners laser 3D. Nous avons vu (Section 2.4) que différentes caractéristiques provenant du modèle peuvent être utilisées (information photométrique, informations géométriques) pour générer les images virtuelles et que différents critères de similarité peuvent être employés pour comparer ces images. Une première étude vise à déterminer la fonction de coût la mieux adaptée à notre cas d'étude pour un calcul de pose dense.

2.4.1 Étude de fonctions de coût

Cette étude a pour objectif de déterminer quelle caractéristique de notre modèle et quel critère de similarité sont les plus intéressants à utiliser pour un calcul de pose par asservissement visuel virtuel dense.

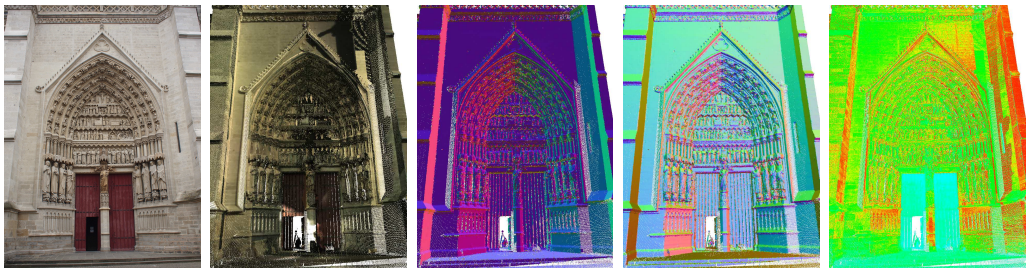
Comme le montrent les méthodes présentées dans la section 2.4, le critère de similarité généralement utilisé en asservissement visuel virtuel dense est l'information mutuelle. Ce choix se justifie principalement par les différences visuelles d'une scène perçue par le capteur de vision à l'origine de l'image numérique désirée et telle qu'elle est représentée dans les images virtuelles de cette même scène. Évidemment, ces différences visuelles sont intrinsèquement liées au modèle 3D utilisé pour générer les images virtuelles. Nous listons ici les caractéristiques du modèle pouvant engendrer des différences visuelles entre l'image d'une scène réelle et l'image d'une scène virtuelle qui nécessiteraient d'utiliser l'information mutuelle pour les comparer :

— Géométrie approximative du modèle
— Textures/couleurs photométriques approximatives du modèle
— Utilisation d'informations autres que photométriques (normales, réflexion, réflectance...)

Notre modèle est composé de mesures 3D provenant de scanners laser et la grande précision de ces appareils nous assure une géométrie virtuelle de la scène très précise et fidèle à la réalité. Chaque mesure 3D possède une couleur provenant d'une image numérique de la scène prise par le scanner. Une image numérique peut être considérée comme l'acquisition visuelle en 2 dimensions d'une scène à un instant donné, similairement, un nuage de points généré à partir d'une station peut être considéré comme l'acquisition géométrique et visuelle en trois dimensions de cette scène à un instant donné. Comme pour l'image numérique, l'aspect visuel du nuage de points dépend directement de la position du scanner et des conditions extérieures (éclairage, éléments mobiles, etc...) de la scène scannée. Par conséquent, l'information photométrique de chaque pixel d'une image virtuelle contenant la projection d'un point 3D du modèle représente l'aspect visuel de la scène réelle en ce point tel qu'il était perçu par la caméra du scanner lors de sa mesure.

Est-il envisageable d'estimer la pose de la caméra à l'origine d'une image numérique désirée par asservissement visuel virtuel dense en utilisant directement l'information photométrique des points 3D du modèle ? Est-il préférable de ne pas utiliser l'information photométrique du modèle mais des caractéristiques géométriques et de comparer les images virtuelles obtenues avec l'image désirée via l'information mutuelle ? Pour répondre à ces questions, nous traçons différentes fonctions de coût selon plusieurs couples de degrés de liberté (ddl) autour d'une pose "solution" et selon différents critères de similarité. La figure 2.7a montre

l'image numérique utilisée pour cette étude. Les figures 2.7(b-e) montrent, respectivement, une image virtuelle générée dans le modèle photométrique, dans le modèle des normales, dans le modèle des réflexions et dans le modèle des réflectances. Ces quatre images virtuelles ont été générées à la même pose "solution" que la pose de la caméra à l'origine de l'image numérique désirée.



(a) (b) (c) (d) (e)

FIGURE 2.7: Images réelle et virtuelles d'une même scène : image numérique désirée (a), image virtuelle du modèle photométrique (b), image virtuelle du modèle des normales (c), image virtuelle du modèle des réflexions (d) et image virtuelle du modèle des réflectances (e)

Les fonctions de coût sont calculées pour les couples de ddl suivants : (Translation en ${}^c\vec{X}$, Translation en ${}^c\vec{Y}$), (Translation en ${}^c\vec{X}$, Rotation autour de ${}^c\vec{Y}$) et (Translation en ${}^c\vec{Z}$, Rotation autour de ${}^c\vec{Z}$).

2.4.1.1 Analyse des fonctions de coût dans le modèle photométrique

Le tableau 2.1 contient les fonctions de coût calculées dans le modèle photométrique. Pour ce modèle, les critères de similarité suivants sont comparés : somme des différences au carré (SSD), somme des différences au carré et la corrélation croisée sur les images centrées et normalisées (ZNSSD et ZNCC) ainsi que l'information mutuelle (IM). Afin de faciliter les comparaisons, nous traçons l'opposé de la ZNCC et de l'IM.

Centrer et normaliser les intensités des images virtuelles et de l'image numérique a une influence importante sur la forme des fonctions de coût. Centrer et normaliser les intensités de deux images que l'on souhaite comparer est connu pour rendre la comparaison robuste aux changements globaux d'illumination entre ces deux images. C'est pourquoi, les fonctions de coût obtenues à partir de la ZNSSD et de la ZNCC ont une forme plus convexe et un minimum plus prononcé comparé à celles obtenues à partir de la SSD. Les fonctions de coût obtenues à partir de l'IM sont également très bonnes, les minimums semblent légèrement plus marqués que ceux des fonctions calculée avec la ZNSSD et la

ZNCC mais globalement les fonctions de coût obtenues à partir de ces trois critères restent très similaires.

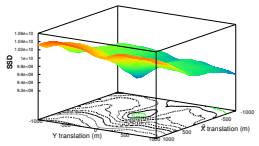
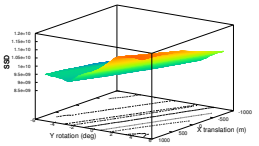
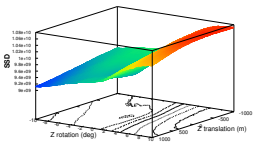
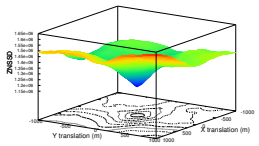
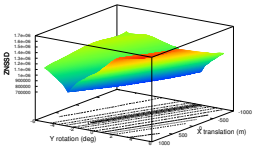
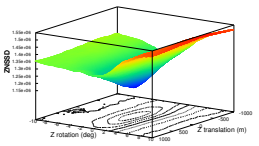
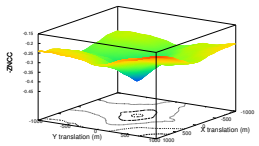
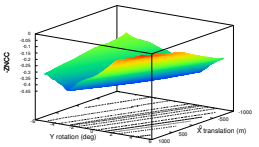
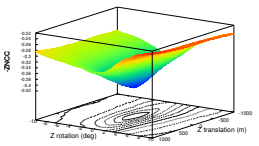
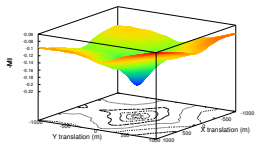
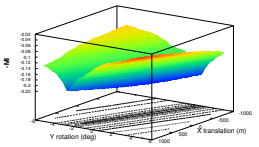
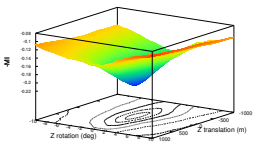
ddl / Similarité	$t({}^c\vec{X})$, $t({}^c\vec{Y})$	$t({}^c\vec{X})$, $R({}^c\vec{Y})$	$t({}^c\vec{Z})$, $R({}^c\vec{Z})$
SSD			
ZNSSD			
ZNCC			
IM			

TABLE 2.1: Étude de fonctions de coût : Modèle photométrique

2.4.1.2 Analyse des fonctions de coût dans le modèle des normales

Le tableau 2.2 contient les fonctions de coût calculées dans le modèle des normales. Les images virtuelles générées dans ce modèle et l'image numérique réelle désirée étant de nature différente, l'information mutuelle est le critère de similarité le plus adéquat et est donc le seul critère utilisé pour calculées les fonctions de coût.

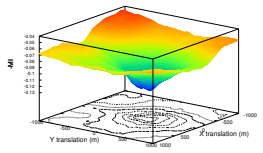
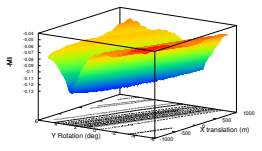
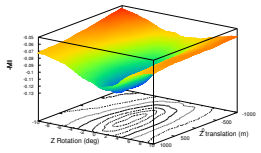
ddl / Similarité	$t({}^c\vec{X})$, $t({}^c\vec{Y})$	$t({}^c\vec{X})$, $R({}^c\vec{Y})$	$t({}^c\vec{Z})$, $R({}^c\vec{Z})$
IM			

TABLE 2.2: Étude de fonctions de coût : Modèle des normales

Les fonctions de coût calculées dans le modèle des normales sont plus chahutées que celles obtenues dans le modèle photométrique. Le minimum global des fonctions est nettement moins bien marqués et n'est pas clairement définis.

2.4.1.3 Analyse des fonctions de coût dans le modèle des réflexions

Le tableau 2.3 contient les fonctions de coût calculées dans le modèles des réflexions. Comme pour le modèle précédent, les images virtuelles et l'image numérique réelle désirée étant de nature différente, l'information mutuelle uniquement est utilisée pour calculer les fonctions de coût.

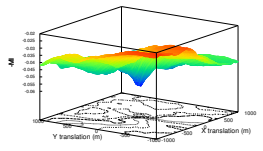
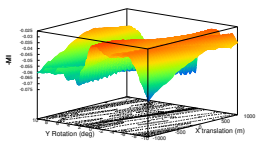
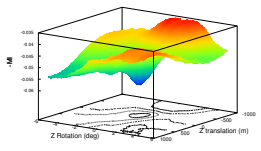
ddl / Similarité	$t({}^c\vec{X})$, $t({}^c\vec{Y})$	$t({}^c\vec{X})$, $R({}^c\vec{Y})$	$t({}^c\vec{Z})$, $R({}^c\vec{Z})$
IM			

TABLE 2.3: Étude de fonctions de coût : Modèle des réflexions

Les fonctions de coût calculées dans le modèle des réflexions ont un minimum global plus marqué et plus identifiable que celui des fonctions obtenues dans le modèle des normales. Cependant, les fonctions sont très chahutés autour de la solution. Par conséquent, l'optimisation pourrait plus facilement tomber dans un minimum local.

2.4.1.4 Analyse des fonctions de coût dans le modèle des réflectances

Le tableau 2.4 contient les fonctions de coût calculées dans le modèle des réflectances. Ici encore, les images virtuelles et l'image numérique réelle désirée sont de nature différente, les fonctions de coût ne sont donc tracés que pour

l'information mutuelle.

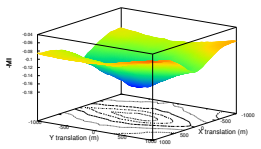
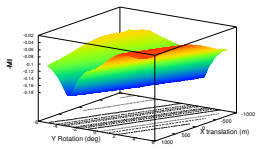
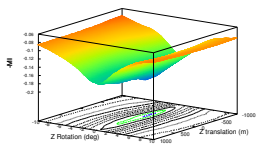
ddl / Similarité	$t({}^c\vec{X})$, $t({}^c\vec{Y})$	$t({}^c\vec{X})$, $R({}^c\vec{Y})$	$t({}^c\vec{Z})$, $R({}^c\vec{Z})$
IM			

TABLE 2.4: Étude de fonctions de coût : Modèle des réflectances

Les fonctions de coût calculées dans le modèle des réflectances sont plus convexes et beaucoup moins chahutées que celles obtenues dans le modèle des normales ou des réflexions. Cependant, elles ne présentent pas un minimum global clairement prononcé et par conséquent difficilement identifiable.

2.4.1.5 Conclusion

Le somme des différences au carré généralement utilisée en asservissement visuel purement photométrique (Section 1.2.2.1) se montre peu recommandable dans notre situation. En effet, cette mesure de similarité est simple et tout a fait efficace pour comparer deux images acquises par une même caméra. Cependant, dans notre cas, les points du modèle photométrique ont une intensité provenant d'une caméra différente de celle utilisé pour obtenir l'image numérique désirée. En ajoutant à cela les problèmes liés à l'assemblage des nuages de point provenant de différentes stations (Section 2.2.1), la SSD montre ses limites. Cependant, le simple fait de centrer et de normaliser les intensités des images permet d'obtenir des fonctions de coût aussi intéressantes que celles obtenues à partir d'une métrique plus complexe comme l'information mutuelle. Les fonctions de coût obtenues à partir des différentes caractéristiques géométriques du modèle sont beaucoup plus bruitées, moins convexes et ont un minimum difficilement atteignables. Toutes ces raisons et observations indiquent qu'il est préférable d'utiliser le modèle photométrique avec comme critère de similarité la ZNSSD, la ZNCC ou l'IM dans le but de faire converger la pose d'une caméra virtuelle vers une pose réelle avec plus de facilité et plus de rapidité. Enfin, la complexité réduite du calcul de la ZNSSD par rapport à la ZNCC ou à l'IM, que ce soit en terme d'erreur à minimiser/maximiser ou en terme de modélisation de la matrice interaction, nous encourage à utiliser la ZNSSD dans la suite de nos travaux.

2.4.2 Calcul de pose sur critère photométrique

Le calcul de pose, dans le cadre d'un asservissement visuel virtuel sur critère photométrique, peut être considéré comme un problème d'optimisation minimisant la différence entre une image numérique désirée $\mathbf{I}$ et les images courantes $\mathbf{Iv}$ générées par une caméra virtuelle évoluant dans un nuage de points coloré par le scanner. La fonction à minimiser est :

$$\mathbf{e} = \widetilde{\mathbf{I}} - \widetilde{\mathbf{Iv}}(\mathbf{r}) \tag{2.6}$$

où $\mathbf{r} = (t_x, t_y, t_z, \theta_{w_x}, \theta_{w_y}, \theta_{w_z})$ représente la pose courante de la caméra virtuelle dans le modèle. Nous verrons dans ce qui suit, qu'à la différence de l'asservissement visuel purement photométrique, l'intégralité des intensités de l'image virtuelle courante et de l'image réelle désirée n'est pas forcément utilisée, c'est pourquoi elles sont notées $\widetilde{\mathbf{I}}$ et $\widetilde{\mathbf{Iv}}(\mathbf{r})$.

Il a été démontré que la commande la plus efficace est basée sur une technique de type Levenberg-Marquardt (LM) [Collewet 2008]. Elle assure une meilleure convergence par rapport à d'autres lois de commande. La loi de commande estimant l'incrément de pose de la caméra virtuelle basée sur LM est :

$$\dot{\mathbf{r}} = -\lambda(\mathbf{H} + \mu diag(\mathbf{H}))^{-1}\mathbf{L}_{\widetilde{\mathbf{Iv}}}^{\mathrm{T}}(\widetilde{\mathbf{Iv}}(\mathbf{r}) - \widetilde{\mathbf{I}}) \tag{2.7}$$

avec $\mathbf{H} = \mathbf{L}_{\widetilde{\mathbf{Iv}}}^{\mathrm{T}}\mathbf{L}_{\widetilde{\mathbf{Iv}}}$ où $\mathbf{L}_{\widetilde{\mathbf{Iv}}}$ est la matrice d'interaction liée aux intensités des pixels utilisés de l'image virtuelle $\mathbf{Iv}$. Cette matrice d'interaction relie les variations des intensités de l'image aux mouvements de la pose de la caméra virtuelle.

2.4.2.1 Contrainte du flot optique

Comme pour l'asservissement visuel purement photométrique (Section 1.2.2.1), la méthode se base sur la contrainte du flot optique. En supposant la scène comme étant Lambertienne, l'intensité I d'un même point $\mathbf{x}$ de l'image virtuelle $\mathbf{I_v}$ reste constante après un mouvement de la caméra :

$$I(\mathbf{x} + d\mathbf{x}, t + dt) = I(\mathbf{x}, t) \tag{2.8}$$

avec $d\mathbf{x}$ le déplacement du point $\mathbf{x}$ dans l'image et dt l'intervalle de temps entre les deux images. Si $\mathbf{x}$ est faible, alors la contrainte du flot optique [Horn 1980] est valide :

$$\vec{\nabla}\mathbf{I}^T\dot{\mathbf{x}} + I_t = 0 \tag{2.9}$$

avec $\vec{\nabla}\mathbf{I}$ le gradient spatial de $I(\mathbf{x}, t)$ et $I_t = \frac{\partial I(\mathbf{x},t)}{\partial t}$ le gradient temporel.

2.4.2.2 Expression de la matrice d'interaction

Les déplacements du point $\mathbf{x} = (x, y)$ dans l'image $\mathbf{Iv}(\mathbf{r})$ sont liés aux mouvements de la caméra virtuelle $\mathbf{v} = \dot{\mathbf{r}}$ par la matrice d'interaction géométrique :

$$\dot{\mathbf{x}} = \mathbf{L}_{\mathbf{x}}\mathbf{v} \tag{2.10}$$

où $\mathbf{v} = (\boldsymbol{\upsilon}, \boldsymbol{\omega})$ contient respectivement les vitesses en translation et en rotation de la caméra virtuelle. La matrice d'interaction géométrique $\mathbf{L}_{\mathbf{x}}$ dépend du modèle de projection de la caméra. Qu'importe le modèle de projection choisi, $\mathbf{L}_{\mathbf{x}}$ nécessite la connaissance d'informations 3D des points de l'image. L'image étant le fruit de la projection d'un modèle virtuel dont la géométrie est fidèle à la réalité, les informations 3D contenues dans la matrice d'interaction ne sont pas approximées.

En substituant l'équation 2.10 dans l'équation 2.9 nous pouvons alors formuler la matrice d'interaction $\mathbf{L}_I(\mathbf{x})$ reliant l'intensité du point $\mathbf{x}$ aux mouvements de la caméra :

$$\mathbf{L}_I(\mathbf{x}) = -\nabla I^T \mathbf{L}_{\mathbf{x}} \tag{2.11}$$

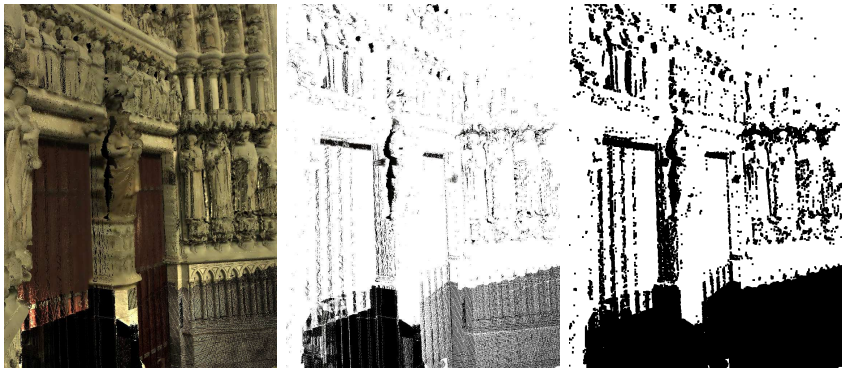
2.4.2.3 Calcul des gradients

Les gradients spatiaux $\vec{\nabla}\mathbf{I}$ de l'image virtuelle $\mathbf{Iv}$ sont les seules données résultantes d'un traitement d'image. Ils sont généralement calculés en considérant un voisinage de taille fixe pour tout point de l'image, généralement deux patchs de taille 1×7 et 7×1 autour de chaque pixel sont utilisés (eq. 1.35 et eq. 1.36). Dans notre cas, $\mathbf{Iv}$ est une image virtuelle, obtenue par la projection d'un nuage de points. Par conséquent, tous les pixels de l'image ne contiennent pas forcément la projection d'un point 3D du modèle et certains pixels sont donc "vides" (Figure 2.8a-b). Prendre en compte ces pixels vides fausserait les gradients de l'image et a fortiori le comportement de la caméra virtuelle durant l'asservissement. C'est pourquoi, nous n'utilisons que les pixels de l'image dont les pixels voisins appartenant aux patchs contiennent la projection d'un point 3D de la scène (Figure 2.8c).

La matrice d'interaction $\mathbf{L}_{\widetilde{\mathbf{I}_\mathbf{v}}}$ est construite en empilant les pixels utiles (Figure 2.8c) qui correspondent à des points 3D du modèle.

$$\mathbf{L}_{\widetilde{\mathbf{I}_\mathbf{v}}} = \begin{bmatrix} \mathbf{L}_{I_0} \\ \mathbf{L}_{I_1} \\ \vdots \\ \mathbf{L}_{I_K} \end{bmatrix} \tag{2.12}$$

où K est égal au nombre de pixels de $\mathbf{Iv}$ utilisables.



(a) (b) (c)

FIGURE 2.8: Pixels utilisés durant l'asservissement : image virtuelle d'un nuage de points (a), pixels contenant la projection d'un point 3D en blanc et pixels vides en noir (b), pixels utilisés pour l'asservissement en blanc et non-utilisés en noir (c)

2.4.2.4 Mise à jour de la pose

La pose de la caméra par rapport à la scène virtuelle ${}^{c}\mathbf{M}_{s}$ est itérativement mise à jour grâce à l'incrémentation de la pose $\dot{r}$ en utilisant l'application exponentielle e du groupe spécial euclidien $se(3)$:

$$ {}^{c^{[t+1]}}\mathbf{M}_{s} = (e^{(\mathbf{r})})^{-1} \times {}^{c^{[t]}}\mathbf{M}_{s} = ({}^{c^{[t]}}\mathbf{M}_{c^{[t+1]}})^{-1} \times {}^{c^{[t]}}\mathbf{M}_{s} = {}^{c^{[t+1]}}\mathbf{M}_{c^{[t]}} \times {}^{c^{[t]}}\mathbf{M}_{s} \quad (2.13) $$

L'optimisation de la pose peut prendre fin selon différents critères d'arrêts choisis en fonction de l'application : un nombre d'itération maximal atteint, une erreur résiduelle constante ou suffisamment basse, etc...

2.5 Applications

Les asservissements visuels virtuels avec un nuage de points 3D comme représentation de la scène ont été utilisés dans le cadre de deux applications distinctes. Dans un premier temps, nous souhaitons retrouver la pose à partir de laquelle une image numérique a été prise dans le but de projeter les couleurs de cette image sur le nuage de points. Dans la seconde application, il s'agit de localiser un robot mobile équipé d'une caméra omnidirectionnelle tout au long de ses déplacements dans un environnement précédemment scanné.

2.5.1 Colorisation photo-réaliste de nuages de points

2.5.1.1 Principes

Combiner l'information photométrique contenue dans des images numériques à la lasergrammétrie comme le font les lasers scanner permet d'obtenir un nuage de points 3D dont la géométrie est très fidèle à la scène réelle mais dont la qualité photométrique ne représente pas parfaitement la réalité. Il apparaît alors intéressant d'utiliser le scanner laser et l'appareil photo numérique séparément. En effet, les mesures du scanner peuvent être combinées aux méthodes de photogrammétrie rapprochée ou à des images numériques pour améliorer la qualité et la résolution de la texture des nuages de points (la texture du nuage correspond ici à l'aspect lié à la couleur des points) avec un photo-réalisme accru. Cela nécessite de recaler les données collectées par les deux appareils (le scanner laser et appareil photo numérique) puisqu'ils n'ont pas de repère commun a priori. Ce recalage est une étape cruciale pour une bonne colorisation, c'est pourquoi ce problème a longuement été étudié et de nombreuses propositions ont été faites. Dans la grande majorité des cas, le modèle 3D n'est pas un nuage de points mais un maillage 3D. Ces approches ont été catégorisées dans [Corsini 2009] et sont brièvement rappelées ici.

— Méthodes poses fixes et connues :
Cette catégorie contient les méthodes les plus simples. Lorsque les poses des deux appareils sont connues relativement l'une par rapport à l'autre, le passage entre les deux systèmes de coordonnées est facilement réalisé puisque la transformation reliant leurs repère est connue [Abmayr 2004].

— Méthodes basées primitives visuelles :
Ces méthodes sont les plus répandues. Elles se basent sur un appariement de primitives visuelles entre les deux jeux de données. Ces primitives visuelles peuvent être des points [Moussa 2012], des droites [Alshawabkeh 2004] ou encore d'autres caractéristiques. Elles peuvent être sélectionnées manuellement [Adan 2012] ou détectées automatiquement [Mastin 2009].

— Méthodes basées silhouettes :
Le recalage peut être obtenu en comparant les formes du modèle 3D avec les contours et l'objet détectés dans l'image numérique désirée [Belkhouche 2012].

— Méthodes basées multi-vues :
Dans ces méthodes, plusieurs images numériques sont acquises afin de générer un nuage de points 3D épars, en Structure from Motion par exemple [Corsini 2012]. Ce nuage de points est recalé avec le modèle 3D ciblé. Une fois ce recalage 3D/3D effectué, les images numériques utilisées pour créer le modèle épars peuvent être alignées avec le modèle 3D et utilisées pour la colorisation.

— Méthodes statistiques :

Le critère de similarité généralement utilisé dans ces méthodes est l'information mutuelle (IM). L'IM est maximisée entre l'image numérique désirée et des caractéristiques géométriques du modèle 3D. Différentes caractéristiques comme les normales, la réflexion ou les occultations ambiantes sont comparées dans [Corsini 2009].

Nous proposons de formaliser ce problème de recalage entre les données issues du scanner et de l'appareil photo numérique comme une estimation de pose sur critère photométrique (Section 2.4). Calculer la pose exacte de la caméra à l'origine d'une image numérique en utilisant une représentation virtuelle précise de l'environnement revient à recaler le modèle virtuel sur l'image numérique, et inversement.

2.5.1.2 Méthodologie

Soit $\mathbf{I}$, une image numérique acquise à une pose quelconque dans un environnement réel ayant été préalablement scanné. Le nuage de points représentant cet environnement est **PCum** et $\mathbf{I_v}(\mathbf{r})$ est une image virtuelle générée à partir d'une pose $\mathbf{r}$ dans **PCum**. Nous cherchons alors la pose $\mathbf{r}$ permettant de recaler photométriquement $\mathbf{I}$ et $\mathbf{I_v}(\mathbf{r})$. Pour cela, nous estimons une première pose de caméra virtuelle approximative en utilisant le calcul de pose basé points (Section 2.3). Ensuite, cette pose est affinée à l'aide du calcul de pose photométrique (Section 2.4).

Estimation de la pose approximative - Des points d'intérêt dans l'image numérique et dans une image virtuelle sont automatiquement détectés et mis en correspondance à l'aide de l'algorithme ASIFT [Morel 2009]. Encore une fois, cette première étape ne pourrait être réalisée sans l'homogénéisation des couleurs des nuages de points composant **PCum** pour les raisons énoncées précédemment.

L'algorithme ASIFT nous fournit donc un jeu d'appariements 2D/2D entre $\mathbf{I}$ et $\mathbf{I_v}(\mathbf{r})$. À partir de la projection inverse de la caméra virtuelle, nous sommes en mesure de remonter aux coordonnées 3D des points ASIFT détectés dans $\mathbf{I_v}(\mathbf{r})$. Nous obtenons ainsi des appariements 2D/3D pouvant être utilisés comme primitives visuelles pour un asservissement visuel virtuel basé points (Section 2.3).

La pose de la caméra virtuelle ainsi obtenue est plus proche de la pose de la caméra à l'origine de l'image numérique réelle. Par conséquent, le contenu de l'image virtuelle générée à cette nouvelle pose est plus proche du contenu de l'image numérique. Il apparaît donc intéressant de réitérer le même procédé de détection, appariement et calcul de pose avec cette nouvelle image virtuelle pour améliorer le recalage. Expérimentalement, il s'avère que réitérer trois fois

de suite le procédé permet d'obtenir une pose de caméra virtuelle satisfaisante pour initialiser le calcul de pose photométrique.

Optimisation photométrique de la pose - Le calcul de pose basé points permet d'obtenir une pose de caméra approximative, relativement proche de la pose de la caméra réelle à l'origine de l'image numérique **I**. Nous exprimons la correction de cette approximation comme un problème d'optimisation dans le cadre d'un asservissement visuel virtuel sur critère photométrique (Section 2.4) avec comme représentation de la scène le modèle **PCum**.

Si la caméra à l'origine de l'image numérique désirée a été calibrée, alors les paramètres intrinsèques de la caméra virtuelle sont simplement fixés comme étant égaux à ceux de la caméra réelle. Si ce n'est pas le cas, alors les paramètres intrinsèques de la caméra virtuelle sont optimisés au même titre que les paramètres extrinsèques durant l'asservissement.

Colorisation - Généralement, la colorisation ou la texturisation est réalisée lorsque le nuage de points est maillé, sur un modèle 3D représentant les surfaces de l'environnement. Pour être en mesure de coloriser directement le nuage de points, il est indispensable de prendre en considération la gestion des occultations du nuage. En effet, certains points provenant de certaines parties de la scène qui sont invisibles d'une position de caméra peuvent être considérés comme visibles et colorisés. Ce phénomène peut se produire dans deux situations : lorsque deux (ou plusieurs) points de différentes profondeurs sont sur la même ligne de vue et sont donc projetés dans un même pixel ou lorsque tous les points, projetés dans un pixel, proviennent de parties invisibles de la scène. Dans les deux cas, des points sont colorisés avec une couleur RVB de l'image numérique qui, en réalité, n'est pas censée leur être attribuée. Pour être en mesure de ne colorer que les points 3D visibles à partir de la pose de la caméra optimisée, la visibilité des points du nuage est estimée avec la méthode "Hidden Point Removal" [Katz 2007]. Les points 3D visibles peuvent alors être colorés avec la couleur des pixels de l'image numérique dans lequel ils sont projetés.

Généralement, plusieurs images numériques sont nécessaires pour coloriser une scène. Dans ce cas, les poses des caméras à l'origine des différentes images numériques sont calculées successivement avec notre approche. Pour la colorisation, les points visibles dans une première image numérique sont colorisés et les autres sont marqués comme non-colorés. Les points visibles dans la deuxième image numérique et marqués comme étant non-colorés sont alors colorisés. L'opération est répétée sur l'ensemble des images numériques ou jusqu'à ce que la totalité des points du nuage soit colorisée.

FIGURE 2.9: Images numériques du portail sud de la cathédrale d'Amiens utilisées pour la colorisation du portail sud de la cathédrale.

2.5.1.3 Résultats qualitatifs

Le portail Sud - En appliquant cette méthode sur le portail de la Vierge dorée de la cathédrale d'Amiens à partir des trois images numériques (Figure 2.9), nous avons obtenu un modèle 3D très réaliste du porche et de ses sculptures environnantes (Figure 2.10b). Depuis, ce modèle a été utilisé dans la conception de jeux sérieux éducatifs [Caron 2013, Leclet-Groux 2013] et pour illustrer une nouvelle méthode d'assistance à la navigation dans les environnements 3D complexes [Habibi 2014].

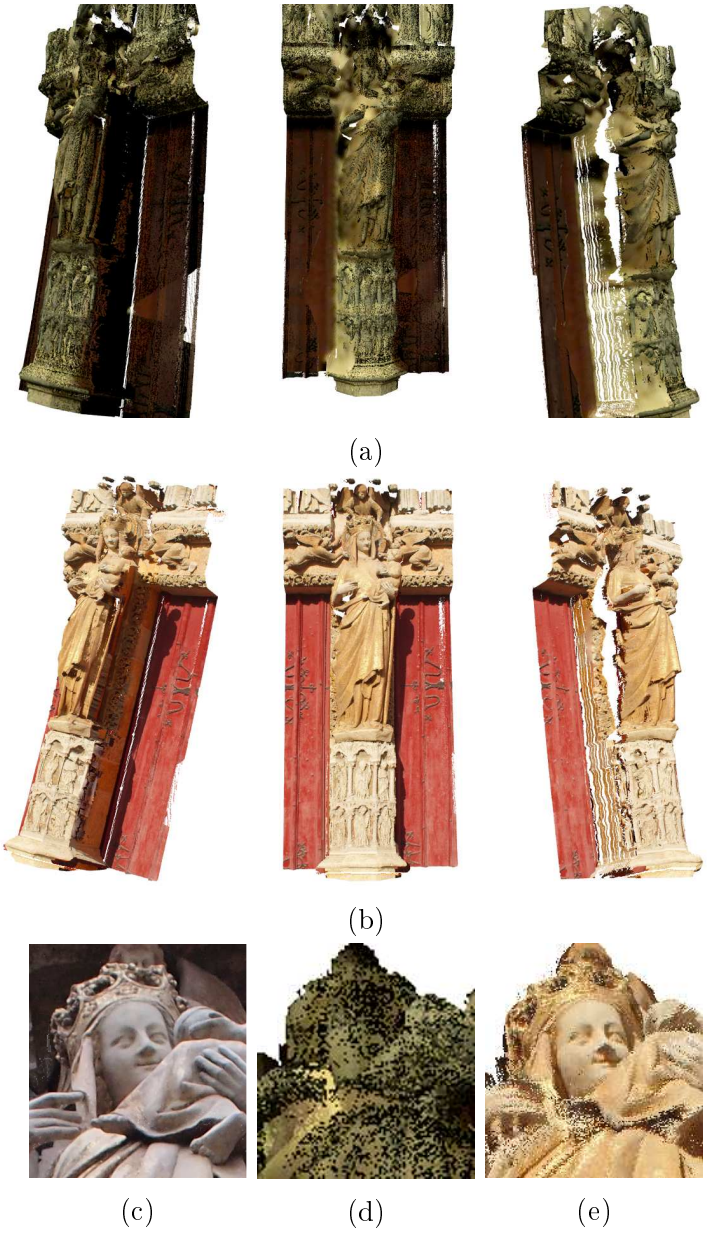
(a) (b)

FIGURE 2.10: Résultat de colorisation : nuages de points du portail Sud avant colorisation (a), et après (b)

FIGURE 2.11: Images numériques de la Vierge Marie utilisées pour la colorisation de la statue

La vierge dorée - Nous nous focalisons sur une plus petite partie du portail sud : la statue de la Vierge Marie. Pour parvenir à un résultat optimal, l'idéal serait que chaque point 3D visible à partir d'une pose de caméra soit projeté dans un pixel différent de l'image numérique. Connaissant les caractéristiques intrinsèques de l'appareil photo et la résolution moyenne du nuage de points, nous avons calculé la distance nécessaire entre la statue et l'appareil photo pour s'assurer qu'un pixel de l'image numérique ne reçoive qu'un unique point 3D projeté. Cette distance a été estimée à quatre mètres en considérant une scène plane fronto-parallèle à l'APN. La figure 2.11 montre trois images numériques prises autour de la statue à environ quatre mètres de distance.

La figure 2.12a montre le nuage de points de la statue dont les couleurs sont celles acquises par le scanner. La figure 2.12b quant à elle montre le nuage de points de la statue après la colorisation. Une image numérique a été prise mais n'a pas été utilisée pour la colorisation. La figure 2.12c montre un agrandissement d'une partie de cette image numérique et les figures 2.12(d-e) montrent des images virtuelles de cette même partie respectivement avant et après la colorisation. Les images numériques utilisées pour la colorisation et celle utilisée pour l'évaluation qualitative ont été prises avec cinq mois d'intervalle (différentes saisons). C'est ce qui explique les différences de colorimétrie entre l'image de la figure 2.12c et celle de la figure 2.12e.



(a) (b) (c) (d) (e)

FIGURE 2.12: Résultat de colorisation : nuages de points de la statue de la Vierge Marie avant colorisation (a), et après (b). Évaluation qualitative : zoom d'une image numérique non-utilisée pour la colorisation (c), le nuage de points avant colorisation (d), et après (e)

La chapelle Saint Sébastien - La méthode a également été utilisée sur un nuage de points de la chapelle Saint Sébastien en tirant profit d'une seule image numérique. La figure 2.13 montre une comparaison entre le nuage de points aux couleurs prélevées par le scanner (partie basse) et après avoir été colorisé avec notre approche (partie haute).

FIGURE 2.13: Nuage de points de la Chapelle Saint Sebastien : les points de la partie basse ont les couleurs prélevées par le scanner, ceux de la partie haute ont été colorisés

2.5.2 Localisation de robot mobile

2.5.2.1 Principes

L'asservissement visuel virtuel photométrique est appliqué dans une application de localisation d'un robot mobile évoluant dans un environnement ayant été préalablement scanné.

Le robot mobile (Figure 2.14a) embarque une caméra omnidirectionnelle calibrée (Figure 2.14b) dont l'axe optique est orienté verticalement. Chaque image numérique omnidirectionnelle acquise par le robot est soumise à un calcul de pose photométrique (Section 2.4), les images virtuelles sont générées à partir d'une caméra virtuelle omnidirectionnelle évoluant dans le nuage de points 3D représentant l'environnement. Estimer la pose de chaque image acquise par le

(a) (b) (c)

FIGURE 2.14: Équipement : Robot mobile Pioneer 3-AT (a) équipé d'une caméra omnidirectionnelle (b) et d'un laser mono-nappe SICK LMS-200 (c)

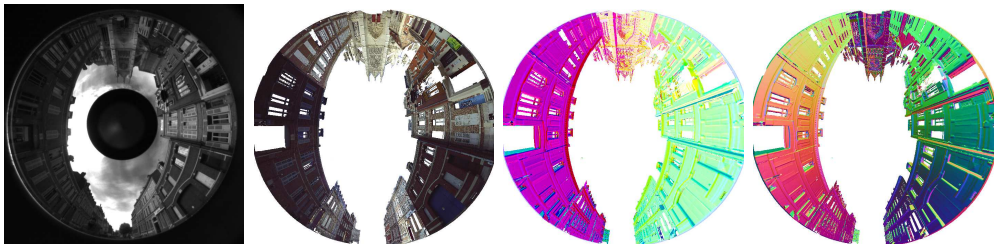
robot revient à suivre sa progression au cours du temps. La toute première pose du robot est considérée comme connue. Pour le calcul de pose des images suivantes, la pose de la caméra virtuelle est initialisée à la pose optimale calculée pour l'image acquise précédemment.

Pour être en mesure de comparer la trajectoire du robot obtenue via notre approche, le robot dispose d'un laser mono-nappe SICK LMS-200 (Figure 2.14c) orienté vers l'avant afin d'effectuer, indépendamment de l'asservissement visuel virtuel photométrique, de la cartographie et de la localisation en simultanées (SLAM).

2.5.2.2 Étude de fonction de coût

Cette partie vise à déterminer si l'utilisation de l'information photométrique du modèle reste un choix intéressant compte tenu du large champ de vue de la caméra utilisée dans cette application. Comme précédemment (Section 2.4.1), des comparaisons de fonctions coût sont menées en utilisant différents critères de similarité, l'information photométrique ou des caractéristiques géométriques du modèle 3D pour représenter l'environnement dans lequel évolue le robot mobile. La figure 2.15a montre l'image numérique omnidirectionnelle utilisée pour cette étude. Les figures 2.15b, 2.15c et 2.15d montrent respectivement une image virtuelle omnidirectionnelle générée dans modèle photométrique, dans le modèle des normales et dans le modèle des réflexions. Ces trois images virtuelles ont été générées à la même pose de caméra que la pose à l'origine de l'image numérique.

Nous comparons les fonctions de coût obtenues en calculant la SSD, la ZNSSD, la ZNCC et l'IM entre l'image numérique (Figure 2.15a) et les images virtuelles



(a) (b) (c) (d)

FIGURE 2.15: Images réelle et virtuelles d'une même scène : image numérique désirée (a), image virtuelle du modèle photométrique (b), image virtuelle du modèle des normales (c), image virtuelle du modèle des réflexions (d)

du modèle photométrique (Figure 2.15b) selon trois couples de degrés de liberté. En ce qui concerne les modèles géométriques des normales (Figure 2.15c) et des réflexions (Figure 2.15d), étant donné la différence de modalité entre l'image réelle et les images virtuelles générées dans ces modèles, nous ne traçons que les fonctions de coût obtenues en calculant l'IM selon les mêmes couples de degrés de liberté. Afin de faciliter les comparaisons, nous traçons l'opposé de la ZNCC et de l'information mutuelle. Contrairement à l'étude réalisée en Section 2.4.1, nous ne montrons ici que quelques fonctions de coût afin de faciliter la lecture.

Nous pouvons observer que, comme pour les images perspectives, les fonctions de coût calculées à partir de la ZNSSD, la ZNCC entre l'image numérique désirée et les images virtuelles générées dans le modèle photométrique ont une forme plus convexe et un minimum plus prononcé que les autres. Les fonctions de coût de l'IM basé sur les caractéristiques géométriques du modèle sont plus bruitées ce qui rendrait difficile la convergence de la caméra virtuelle vers la pose désirée. Ici aussi, le fait de centrer et normaliser les intensités de l'image numérique et de l'image virtuelle permet de faire face aux problèmes liés à la fusion des nuages de points provenant de stations différentes. L'erreur à minimiser étant calculée sur ces intensités, la forme des fonctions de coût obtenues à partir de la ZNSSD et de la ZNCC est plus lisse, convexe et présente un minimum clairement identifiable par rapport à celle obtenu à partir de la SSD. Ces observations indiquent qu'il est préférable d'utiliser la ZNSSD ou la ZNCC ainsi que le modèle photométrique représentant l'environnement dans lequel le robot évolue. Le calcul de la ZNSSD étant plus rapide que la ZNCC, que ce soit pour l'erreur à minimiser/maximiser ou dans la conception de la matrice d'interaction, nous choisissons d'utiliser la ZNSSD dans cette application.

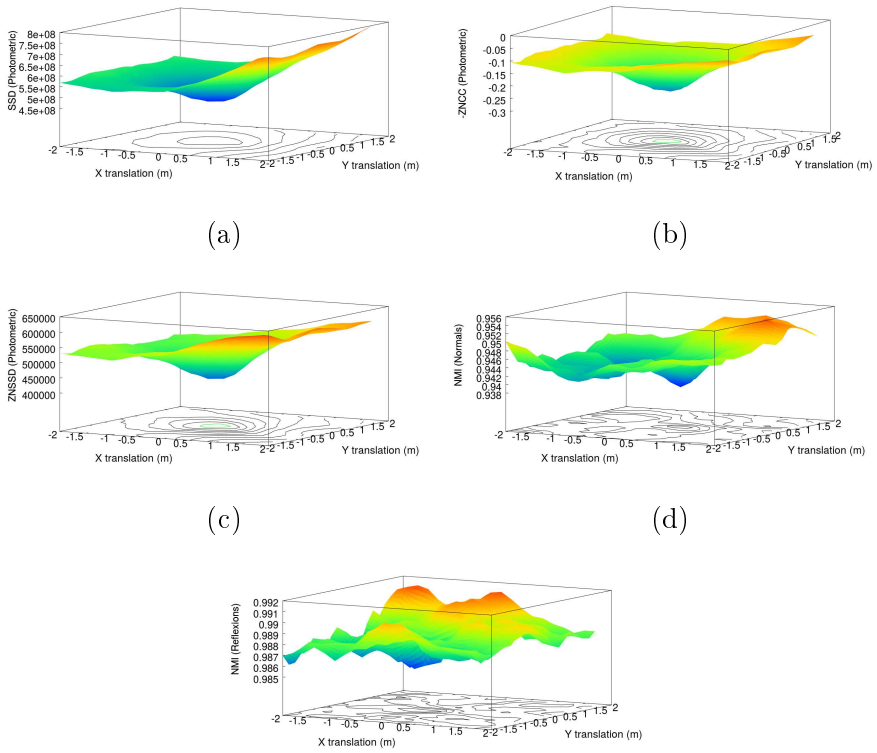


FIGURE 2.16: Étude de fonction de coût : SSD dans le modèle photométrique (a), ZNSSD dans le modèle photométrique (b), ZNCC dans le modèle photométrique (c), IM dans le modèle des normales (d), IM dans le modèle des réflexions (e)

2.5.2.3 Implémentation

Contrairement à la colorisation des nuages de points, les temps de calcul pour une application de localisation de robot mobile doivent être faibles et compatibles avec la commande afin de garder des performances temps-réel.

La génération d'une image virtuelle à partir d'un modèle constitué d'un grand nombre de points 3D est une opération pouvant demander un temps très important. Pour le réduire, la structuration spatiale du modèle 3D (Section 2.2.2) est alors très importante pour être capable de générer rapidement une image virtuelle sans perte d'informations visuelles de l'environnement. Au début de l'expérimentation, l'ensemble des nuages de points organisés de la base de données est chargé. Pendant les déplacements du robot, le nuage de points organisé utilisé est celui qui a été créé à partir de la pose la plus proche de la pose optimale du robot précédemment calculée.

Les différentes étapes du processus de localisation (génération de l'image vir-

tuelle, calcul des gradients, de la matrice d'interaction, du vecteur d'erreur et du vecteur de mise jour de la pose) ont été parallélisées et sont calculées sur processeur graphique (GPU). La parallélisation massive offerte par ces architectures récentes est de plus en plus exploitée pour des opérations qu'elles soient graphiques ou non.

Grâce à la structuration spatiale du modèle et à l'implémentation GPU du calcul de pose, nous sommes en mesure d'estimer la pose du robot à une fréquence de 4Hz en utilisant des images de taille 350×350.

2.5.2.4 Résultats

Deux expérimentations ont été menées dans des environnements différents et en utilisant deux types de caméras grand angle. La première se déroule en extérieur, dans des rues bordants la cathédrale d'Amiens, avec une caméra omnidirectionnelle et la seconde en intérieur, dans la cathédrale d'Amiens, avec une caméra fisheye.

Expérimentation en extérieur :

L'environnement dans lequel évolue le robot est composé de quatre rues formant un parcours fermé de quatre cent mètres. Un modèle 3D des quatre rues (Figure 2.20) a été obtenu par lasergrammétrie en plaçant le scanner laser à treize positions géographiques. Le modèle résultant de ces acquisitions contient plus de dix millions de points 3D, ses couleurs ont été homogénéisées et une base de données de nuages de points organisés a été générée.

Le but de l'expérimentation est d'estimer la pose 3D de la caméra de chaque image acquise par le robot se déplaçant dans les rues du modèle. La vitesse du robot et la fréquence d'acquisition des images permettent de respecter l'équation de contrainte du flot optique (eq. 1.32).

Sur la vue aérienne des quatre rues de la figure 2.18a, le tracé bleu montre la trajectoire estimée par asservissement visuel virtuel sur, approximativement, 21000 images acquises par le robot durant son déplacement, le tracé rouge montre la trajectoire résultant du SLAM obtenue à partir des informations retournées par le laser mono-nappe SICK LMS-200 du robot. La ligne du milieu de la figure 2.18 montre quatre images numériques acquises pendant les déplacements du robot. La position d'acquisition de ces images correspond aux endroits labellisés de (1) à (4) dans la figure 2.18a. La ligne du bas de la figure 2.18 montre les quatre images virtuelles générées aux poses de caméra estimées à partir des quatre images numériques correspondantes.

Il est intéressant de noter que certaines parties de la trajectoire n'ont pas pu être estimées. À plusieurs endroits, le robot doit monter ou descendre du trottoir

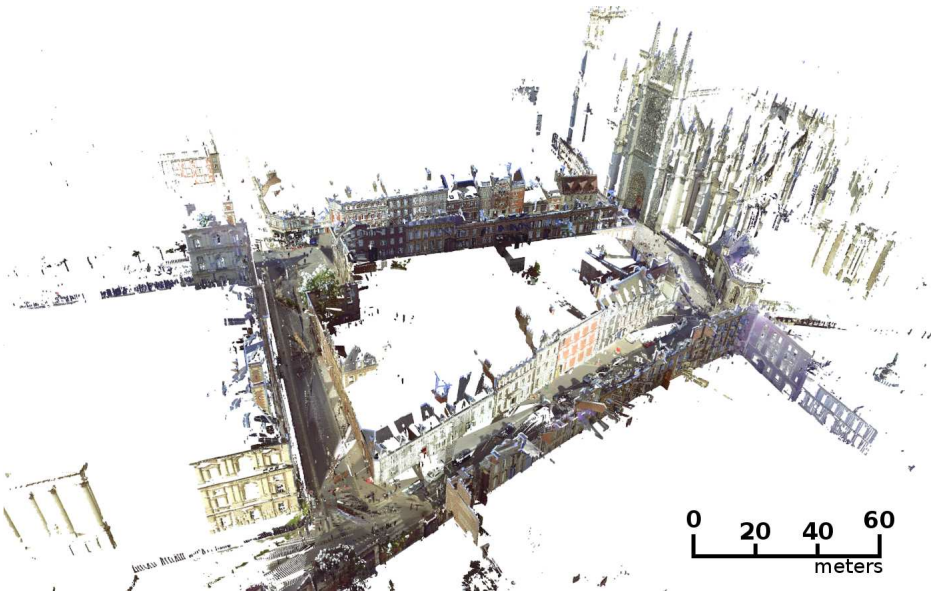


FIGURE 2.17: Modèle 3D de quatre rues composé de treize nuages de points totalisant plus de dix millions de relevés 3D

pour rejoindre l'autre côté de la rue. Durant ces passages, la caméra montée sur le robot est fortement secouée, par conséquent l'écart entre deux images successives peut être trop important pour que la caméra virtuelle parvienne à converger vers la pose de caméra réelle courante.

Le virage entre la deuxième et la troisième rue met en défaut la localisation. Les bâtiments bordant cette partie sont, pour la plupart, soit très éloignés du robot, soit masqués par la présence d'arbres. Ces arbres ont été supprimés du modèle 3D représentant l'environnement. Qui plus est, l'acquisition du modèle par lasergrammétrie et la localisation elle-même se déroulent à plusieurs mois d'intervalle. Pour ces différentes raisons, les informations visuelles perçues par le robot et générées virtuellement sont trop différentes pour être correctement comparées.

Enfin, dans la première rue, le passage de plusieurs véhicules à proximité du robot occulte (Figure 2.19a) complètement les bâtiments clairs de l'autre côté de la rue (Figure 2.19b). Cette différence importante sur une grande partie de l'image ne permet pas à la caméra virtuelle de converger correctement vers la pose de la caméra à l'origine de l'image numérique désirée. L'utilisation d'un estimateur robuste [Comport 2003a] peut régler ce problème d'occultation de la scène mais n'a pas été implémenté dans notre méthode.

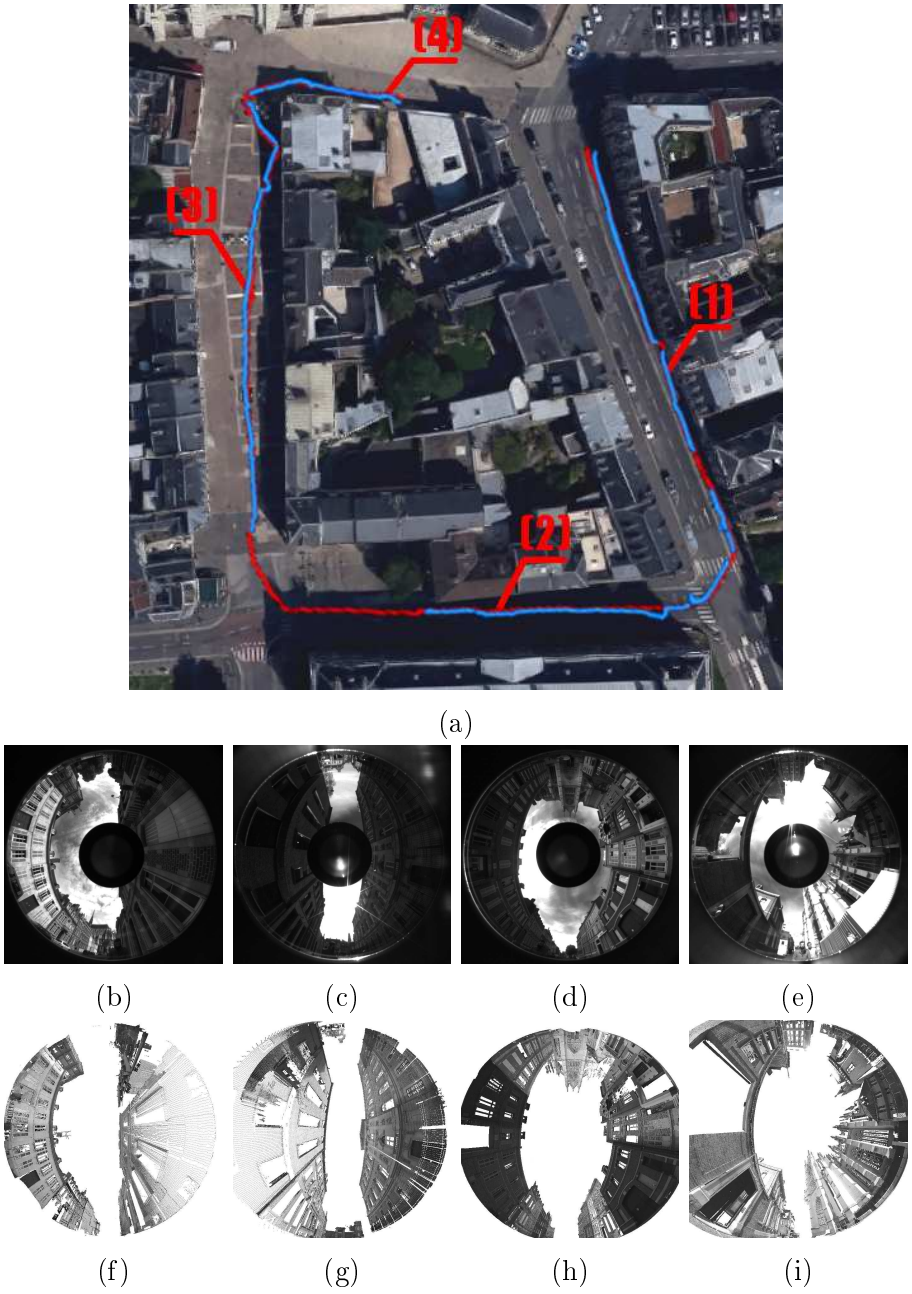
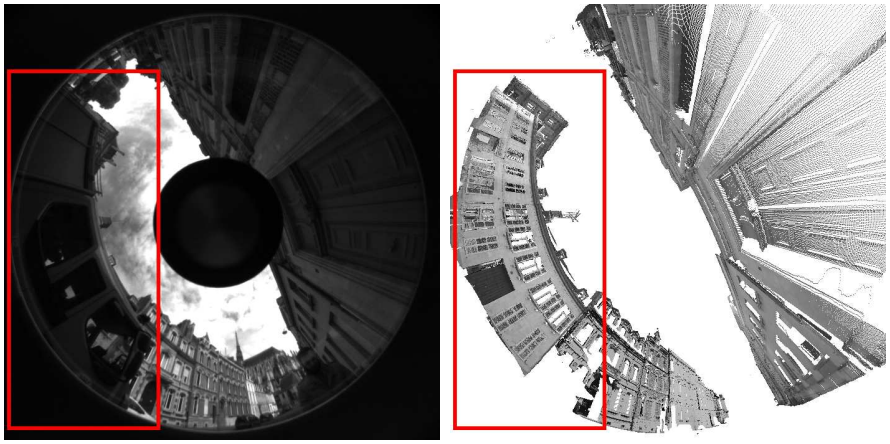


FIGURE 2.18: Vue aérienne des quatre rues (a) : la trajectoire du robot estimée par le SLAM (en rouge) et la trajectoire du robot estimée via asservissement visuel virtuel (en bleu)
Images numériques acquises par le robot dans les quatres rues (b-e). Images virtuelles générées aux poses de caméra éstimées à partir de ces images numériques (f-i). Ces images correspondent aux endroits labelisés (1-4) sur la vue aérienne



(a) (b)

FIGURE 2.19: Occultation d'une partie de l'environnement dans lequel évolue le robot : image numérique réelle, un camion de couleur foncée passant près du robot (a), image virtuelle générée à une pose de caméra correspondant à l'image numérique (b)

Expérimentation en intérieur :

Le même type d'expérimentation a été mené dans la cathédrale d'Amiens (Figure 2.1) en utilisant une caméra fisheye. Nous avons ainsi pu valider notre méthode dans un environnement intérieur avec un capteur de vision différent. L'environnement a été scanné en plaçant le scanner à laser à plus de 50 stations. Le modèle composé des nuages fusionnés compte plus de 30 millions de points 3D (Figure 2.20).

Comme pour l'expérimentation précédente, les couleurs des nuages composant le modèle complet ont été homogénéisées et une base de données de nuages de points organisés a été générée. Une distance de 1 mètre entre les caméras équirectangulaires a été expérimentalement choisie pour créer cette base de données afin de limiter les occultations. Le but de l'expérimentation est d'estimer une trajectoire d'une vingtaine de mètres empruntée par le robot à l'intérieur de la cathédrale à partir d'une séquence d'environ 600 images acquises par sa caméra fisheye. La figure 2.21a montre la trajectoire du robot virtuellement représentée dans le modèle 3D de la cathédrale estimée avec notre approche. La ligne du milieu de la figure 2.21 montre trois images numériques acquises pendant les déplacements du robot. La position d'acquisition de ces images correspond aux endroits labellisés de (1) à (3) dans la figure 2.21a. La ligne du bas de la figure 2.21 montre les trois images virtuelles générées aux poses de caméra estimées à partir des trois images numériques correspondantes.

Les plus grandes difficultés dans cette expérimentation sont les virages. En

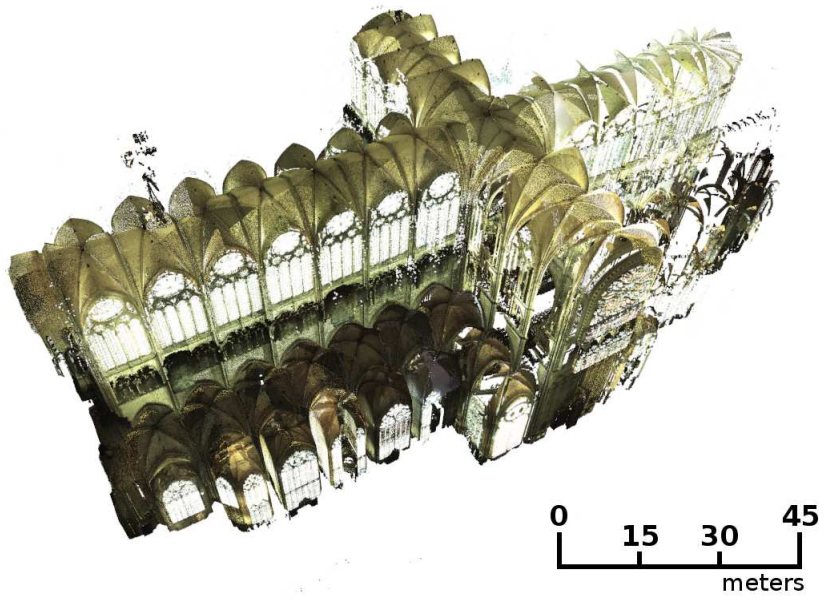
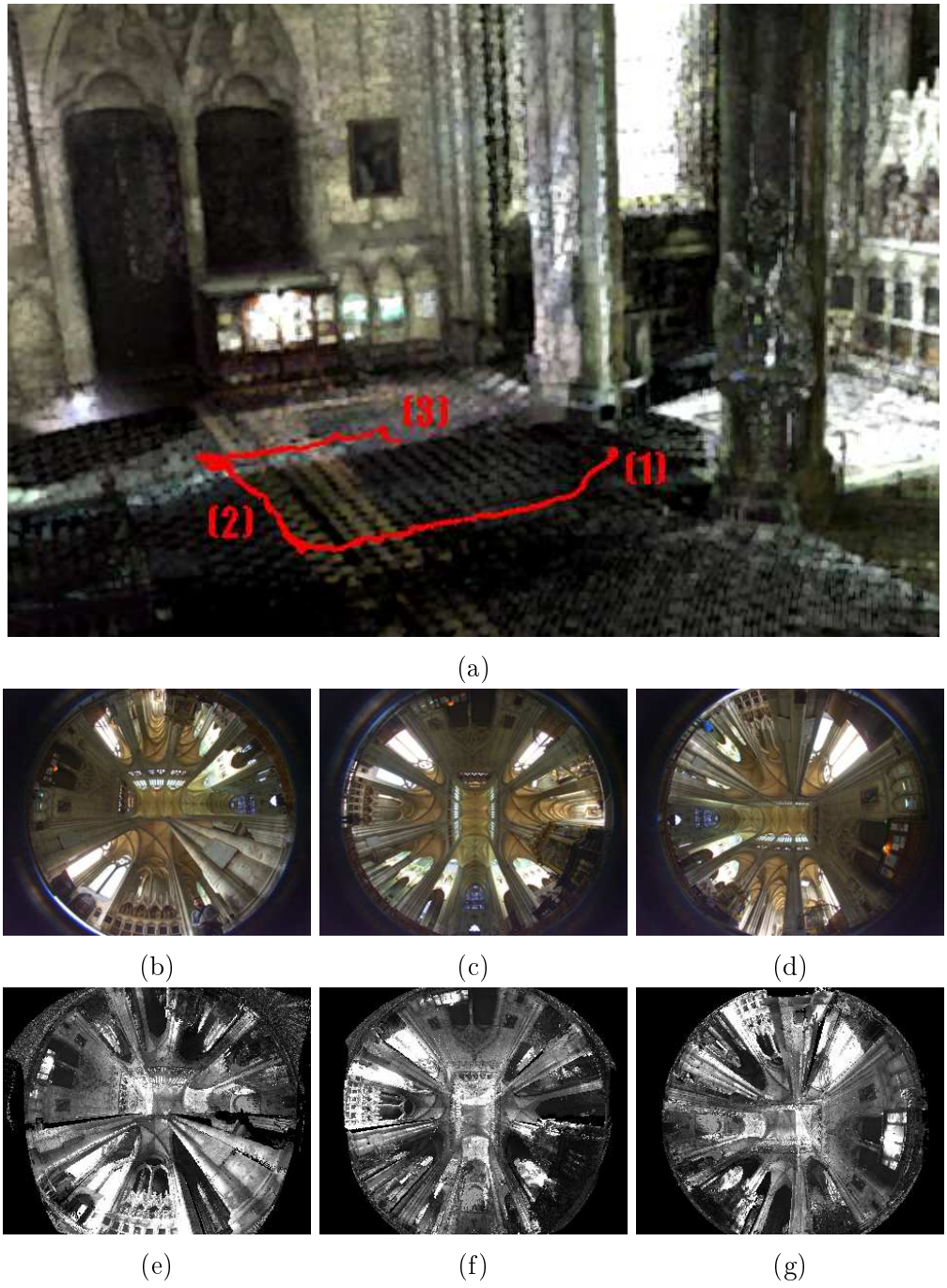


FIGURE 2.20: Modèle virtuel de l'intérieur de la cathédrale d'Amiens composé de plus 30 millions de points 3D

effet, au cours de ses déplacements, le robot effectue de pures rotations autour de son centre de gravité. Durant ces virages, si la vitesse de rotation du robot est trop importante, l'image numérique courante désirée peut être "photométriquement" très éloignée de l'image virtuelle générée à partir de la précédente pose optimale estimée. Ces rotations importante du robot engendre une rotation de la caméra autour de son axe optique difficile à corriger par asservissement visuel.



(a)

(b) (c) (d)

(e) (f) (g)

FIGURE 2.21: Expérimentation en intérieur : la trajectoire du robot estimée via asservissement visuel virtuel, images numériques acquises par le robot (b-d), images virtuelles générées aux poses de caméra éstimées à partir de ces images numériques (e-f). Ces images correspondent aux endroits labelisés (1-3) sur la trajectoire

2.6 Conclusion

Dans ce chapitre, nous avons montré qu'il est possible de résoudre le problème d'estimation de pose de caméra sous le formalisme de l'asservissement visuel virtuel en utilisant directement les caractéristiques photométriques d'images réelles et virtuelles. Nos travaux s'inscrivent dans le programme de recherche E-Cathédrale, c'est pourquoi, la représentation virtuelle de notre environnement de travail est un modèle 3D composé d'un assemblage de plusieurs nuages de points colorés. Aucun des travaux d'asservissement visuel virtuel de l'état de l'art n'exploite directement les intensités de l'image, ni d'environnement 3D sous forme de nuage de points colorés.

Dans un premier temps, nous avons proposé des méthodes de pré-traitement des nuages de points colorés composant le modèle complet. Les modèles provenant d'acquisitions par scanners laser peuvent contenir un nombre gigantesque de mesures 3D. Pour permettre l'exploitation du modèle dans des conditions acceptables, nous avons proposé une structuration spatiale du modèle. Cette structuration permet de n'utiliser que les points utiles du modèle en fonction de la position de la caméra pour créer les images virtuelles. Les modèles provenant d'acquisitions par scanners laser sont, généralement, composés de plusieurs nuages de points acquis à des moments différents de la journée. Les couleurs des nuages de points ont donc des teintes différentes, ce qui génère des incohérences visuelles. L'homogénéisation des couleurs des nuages de points permet de corriger l'aspect visuel du modèle et, par extension, des images virtuelles générées dans ce modèle.

L'homogénéisation des couleurs rend possible la détection et la mise en correspondance de primitives géométriques entre les images virtuelles et les images réelles de la scène. Des calculs de pose à l'aide de l'asservissement visuel virtuel basé points sont alors réalisables et nous proposons de les utiliser pour initialiser le calcul de pose photométrique.

Une étude visant à déterminer quel type de modèle et quel critère de similarité sont les plus pertinents a été réalisée. Puis, une formulation de l'asservissement visuel virtuel photométrique avec comme représentation virtuelle de l'environnement un modèle composé de nuages de points colorés a été proposée.

Les différentes expérimentations, que ce soit pour la colorisation du modèle ou pour la localisation de robot mobile, ont permis de valider l'approche et de mettre en évidence les limitations de la méthode. Ces limites sont principalement liées à l'étroitesse du domaine de convergence des asservissements visuels photométriques.

Chapitre 3

Asservissement visuel basé mélanges de gaussiennes photométriques

Sommaire

3.1 Introduction

L'utilisation de toute l'information photométrique contenue dans les images comme primitive visuelle permet de passer outre à la détection, l'appariement ou encore le suivi de primitives locales. Grâce à cette redondance d'informations, les asservissements visuels se basant sur cette primitive ont une excellente précision de convergence. Cependant, en considérant la pose optimale de la caméra comme étant la solution d'un problème d'optimisation non-linéaire, la convergence vers cette solution dépend directement de la distance entre la pose initiale de la caméra et la pose désirée. Concrètement, pour qu'une méthode d'optimisation réussisse à résoudre un asservissement visuel, l'image obtenue à la pose désirée et l'image

obtenue à la pose initiale doivent avoir un taux de recouvrement d'information photométrique suffisamment important pour que la caméra puisse converger vers la solution.

Ce chapitre propose une nouvelle modélisation des images qui permet d'agrandir le domaine de convergence des asservissements visuels photométriques classiques, lorsque l'image désirée et les images acquises durant l'optimisation sont issues d'une même caméra.

3.2 Mélange de gaussiennes

3.2.1 Motivations

D'un point de vue continu, une image optique est un signal généralement représenté par une fonction bidimensionnelle $f(i,j)$ représentant l'image en chaque point (i,j) de son espace (ex : l'intensité). L'image discrète, c'est-à-dire la représentation numérique du signal continu, est obtenue par la discrétisation des coordonnées spatiales de ce signal dans les deux dimensions de l'image, et par la quantification de la fonction continue, la transformation de la mesure de l'énergie lumineuse continue vers une valeur numérique.

La discrétisation spatiale du signal continu est une opération d'échantillonnage classique. Elle est donc obtenue en disposant régulièrement des échantillonneurs sur la surface de l'image. Cet échantillonneur est généralement représenté par une impulsion de Dirac $\delta(i,j)$.

$$\delta(i,j) = \begin{cases} 0, & (i,j) \neq (0,0) \\ 1, & (i,j) = (0,0) \end{cases} . \tag{3.1}$$

L'image discrète est donc le fruit de la convolution du signal continu (Figure 3.1a) avec une brosse de Dirac (Figure 3.1b) de pas (T_i, T_j) fixé, respectant la loi de Shannon. En niveau de gris, chaque point échantillonné est repéré par des coordonnées discrètes $\mathbf{u} = (u,v)$ et possède pour intensité $I(\mathbf{u})$. En considérant que chaque impulsion de la brosse de Dirac ait pour coordonnées (uT_i, vT_j) et soit exprimée par $\delta(uT_i, vT_j)$, l'intensité de ce point échantillonné est :

$$I(\mathbf{u}) = \iint f(i,j)\delta(i - uT_i, j - vT_j)didj \tag{3.2}$$

L'image numérique peut alors être considérée comme une brosse d'impulsions de Dirac dont les amplitudes expriment l'intensité des points échantillonnés (Figure 3.1c).

L'idée générale qui sous-tend ce chapitre est de changer la représentation des pixels pour augmenter ce que l'on peut appeler leur pouvoir d'attraction. En

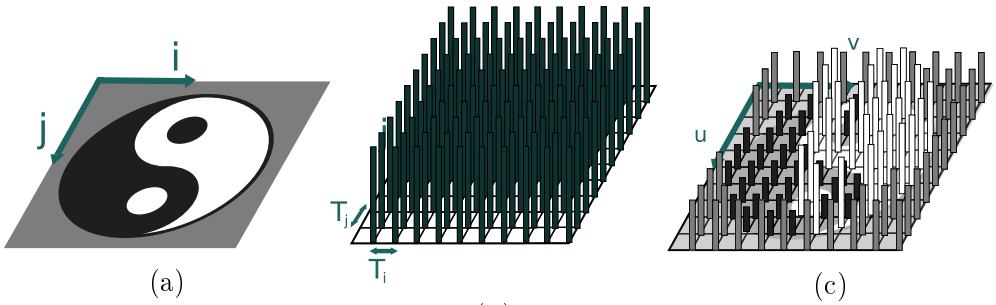


FIGURE 3.1: Discrétisation et quantification d'un signal image : signal continu (a), brosse de Dirac (b), image discrète dont chaque pixel peut être vu comme une impulsion de Dirac d'amplitude égale à l'intensité de ce pixel (c)

considérant que chaque pixel possède un pouvoir d'attraction, lorsqu'un pixel $\mathbf{u} = (u, v)$ est représenté par une impulsion de Dirac (Figure 3.1c), son pouvoir d'attraction est concentré uniquement en $\mathbf{u}$ et est nul partout ailleurs. Il semble alors intéressant de remplacer la fonction de Dirac par une fonction paramétrable ayant une distribution plus étendue de limite nulle en l'infini comme, par exemple, une fonction gaussienne. De plus, la fonction gaussienne est dérivable, ce qui s'avère être très important dans la modélisation de la matrice d'interaction (eq. 3.8) liant l'action et la perception visuelle.

3.2.2 Fonction gaussienne représentant un pixel

Soient deux variables non corrélées $u_g \in \mathbb{R}$, $v_g \in \mathbb{R}$ formant le vecteur $\mathbf{u}_g = (u_g, v_g)$, la gaussienne 2D est exprimée par la fonction de distribution :

$$f(\mathbf{u}_g, \mathbf{u_0}, \boldsymbol{\sigma}) = A \exp\left(-\left(\frac{(u_g - u_0)^2}{2\sigma_u^2} + \frac{(v_g - v_0)^2}{2\sigma_v^2}\right)\right) \tag{3.3}$$

où A est le coefficient d'amplitude, $\mathbf{u_0} = (u_0, v_0)$ est l'espérance mathématique et $\boldsymbol{\sigma} = (\sigma_u, \sigma_v)$ est l'écart-type de la fonction gaussienne. Dans la suite, nous n'utiliserons pas cette terminologie parce que l'utilisation que nous faisons des gaussiennes n'est pas statistique. Aussi, pour éviter les ambiguïtés et les abus de langage, nous appelons $\mathbf{u_0} = (u_0, v_0)$ le centre de la gaussienne et $\boldsymbol{\sigma} = (\sigma_u, \sigma_v)$ son envergure le long de ses axes $\vec{u_g}$ et $\vec{v_g}$.

La figure 3.2 montre un exemple de gaussienne 2D d'amplitude $A = 1$, centrée en $\mathbf{u_0} = (0, 0)$ et d'envergure $\boldsymbol{\sigma} = (0.5, 0.5)$.

Afin d'agrandir le champ d'attraction de chaque pixel de l'image tout en conservant le caractère discriminant des uns par rapport aux autres, nous choisissons de rendre l'envergure des gaussiennes le long des axes $\vec{u_g}$ et $\vec{v_g}$ égales

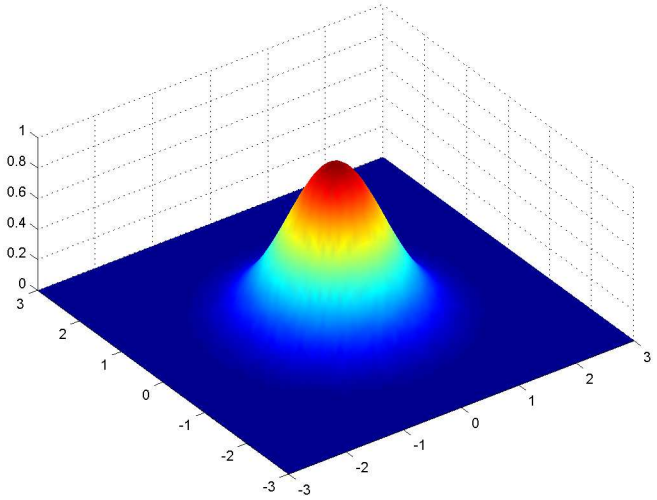


FIGURE 3.2: Exemple de gaussienne 2D d'amplitude $A = 1$, centrée en $\mathbf{u_0} = (0, 0)$ et d'envergure $\boldsymbol{\sigma} = (0.5, 0.5)$

et proportionnelles à l'intensité du pixel qu'elles représentent. Autrement dit, pour un pixel situé aux coordonnées $\mathbf{u} = (u, v)$ d'une image $\mathbf{I}$, l'envergure de la gaussienne représentant ce pixel nous donne : $\sigma_u = \sigma_v = I(\mathbf{u})$. D'un point de vue pratique, durant l'asservissement visuel, ce choix a pour but de créer une "attraction" entre les pixels d'intensité similaire, a fortiori d'envergure de gaussienne similaire. Les pixels noirs ($I(\mathbf{u}) = 0$) ne peuvent être pris en compte car ils entraîneraient une division par 0 dans l'équation (3.3) et sont simplement inutilisés. Nous verrons par la suite qu'il est très intéressant de pouvoir agir sur l'envergure des gaussiennes au cours de l'asservissement. Pour cela, nous ajoutons un paramètre d'extension λ_g servant à pondérer l'envergure $\boldsymbol{\sigma}$ des gaussiennes.

Il parait logique que la gaussienne représentant un pixel soit centrée sur la position de ce pixel dans l'image. En effet, le pouvoir d'attraction attribué au pixel est ainsi à son maximum à la position du pixel et de plus en plus faible à mesure que l'on s'en éloigne dans l'image. Par conséquent, toujours pour un pixel situé aux coordonnées $\mathbf{u} = (u, v)$, le paramètre $\mathbf{u}_g$ de l'équation 3.3 est égal à $\mathbf{u}$.

Enfin, l'amplitude A de toutes les gaussiennes est fixée à 1. Pour résumer, chaque gaussienne est centrée sur le pixel qu'elle représente, avec une amplitude égale à 1 et une envergure dépendante de l'intensité du pixel représenté. La fonction gaussienne photométrique devient alors :

$$g(\mathbf{u}_g, \mathbf{u}) = \exp\left(-\left(\frac{(u_g - u)^2 + (v_g - v)^2}{2\lambda_g^2 I(\mathbf{u})^2}\right)\right) \tag{3.4}$$

où λ_g est le paramètre d'extension de la gaussienne.

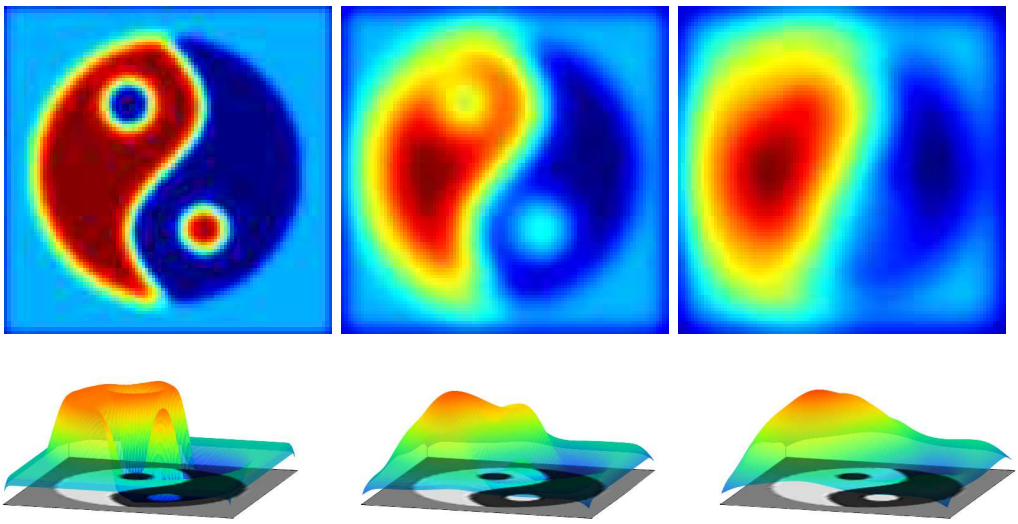
3.2.3 Mélange de gaussiennes d'une image

Chaque pixel d'une image $\mathbf{I}$ est représenté par une fonction gaussienne 2D suivant l'équation (3.4). Pour une image $\mathbf{I}$ de taille $(N \times M)$ nous avons alors un nombre fini $(N \times M)$ (en faisant abstraction des bords) de fonctions gaussiennes de la taille de $\mathbf{I}$. La combinaison de ces $(N \times M)$ fonctions forme un mélange de gaussiennes. La valeur du mélange de gaussiennes d'une image $\mathbf{I}$ en $\mathbf{u}_g$ est la somme des $(N \times M)$ valeurs des gaussiennes en ce point :

$$gm(\mathbf{I}, \mathbf{u}_g) = \sum_{\mathbf{u}} (\alpha_{\mathbf{u}} g(\mathbf{u}_g, \mathbf{u})) \tag{3.5}$$

où chaque pixel de l'image peut être pondéré par $\alpha_{\mathbf{u}}$. Tous les poids sont fixés à 1 pour ne pas favoriser un pixel plutôt qu'un autre. Dans tout ce qui suit, l'échantillonnage spatial d'un mélange de gaussiennes est identique à l'échantillonnage de l'image qu'elle représente, mais nous n'y sommes pas limités.

La figure 3.3 montre des mélanges de gaussiennes de l'image du Yin et du Yang (Figure 3.1c) calculées pour différents paramètres d'extension.



(a) (b) (c)

FIGURE 3.3: Mélanges de gaussiennes de l'image du Yin et du Yang (Figure 3.1c) pour différentes valeurs de λ_g : visualisations 2D et 3D pour $\lambda_g = 0.01$ (a), respectivement pour $\lambda_g = 0.03$ (b) et pour $\lambda_g = 0.05$ (c)

Il est intéressant de noter que pour un faible paramètre d'extension (Figure 3.3a), le mélange de gaussiennes est très proche de l'image d'origine (Figure 3.1c). Par conséquent, l'asservissement visuel basé sur des mélanges de gaussiennes à faibles extensions est similaire à un asservissement visuel purement photométrique. Au contraire, plus le paramètre d'extension est élevé (Figure 3.3b

et Figure 3.3c) et plus les mélanges de gaussiennes s'étendent, favorisant ainsi le chevauchement d'informations nécessaire à la minimisation entre un mélange de gaussiennes désiré et les mélanges de gaussiennes générés au cours de l'asservissement. Ces observations montrent qu'il serait intéressant de commencer l'asservissement visuel avec un paramètre d'extension élevée, puis à convergence, de tendre vers un faible paramètre d'extension. La comparaison des fonctions de coût (Section 3.3.1) confirme ce point de vue.

3.3 Mélanges de gaussiennes comme caractéristiques visuelles denses

Le but de l'asservissement visuel purement photométrique (Section 1.2.2.1) est de minimiser la différence entre les intensités d'une image désirée $\mathbf{I}^*$ et les intensités des images acquises aux cours de l'asservissement $\mathbf{I}(\mathbf{r})$ où $\mathbf{r}$ représente la pose de la caméra. Dans notre cas, nous ne travaillons pas directement sur les images acquises par la caméra mais sur les mélanges de gaussiennes calculés à partir de ces images. Nous notons $\mathbf{gm}(\mathbf{I})$, l'échantillonnage spatial d'un mélange de gaussiennes calculé à partir d'une image $\mathbf{I}$. Par conséquent, la fonction de coût à réguler à 0 est :

$$\mathbf{e} = \mathbf{gm}(\mathbf{I}^*) - \mathbf{gm}(\mathbf{I}(\mathbf{r})) \tag{3.6}$$

où $\mathbf{gm}(\mathbf{I}^*)$ est le mélange de gaussiennes calculé sur l'image désirée avec un paramètre d'extension fixe λ_g^* et $\mathbf{gm}(\mathbf{I}(\mathbf{r}))$ est le mélange de gaussiennes calculé sur l'image acquise à la pose de caméra $\mathbf{r}$ dont le paramètre d'extension λ_g est réévalué au même titre que la pose de la caméra. Le choix du paramètre d'extension désiré λ_g^* et l'initialisation du paramètre d'extension asservi λ_g sont discutés ci-après.

3.3.1 Étude de la fonction de coût

Cette étude a pour but de montrer l'impact de la représentation des images sous formes de mélanges de gaussiennes sur la fonction de coût (eq. 3.6) que l'on cherche à réguler. Pour cela, nous traçons la fonction de coût obtenue en déplaçant une caméra virtuelle selon deux degrés de liberté (translation le long de ${}^c\vec{X}$ et translation le long de ${}^c\vec{Y}$) autour d'une pose désirée. La scène choisie est volontairement minimaliste pour simplifier la lecture des résultats. La figure 3.4a montre l'image virtuelle générée à la pose désirée.

Nous comparons la forme des fonctions de coût obtenues en utilisant différents paramètres d'extension pour calculer les mélanges de gaussiennes (Figures 3.4(c-e)) ainsi que la fonction de coût obtenue en utilisant le critère de similarité

généralement utilisé en asservissement visuel purement photométrique : la somme des carrés des différences (Figure 3.4b).

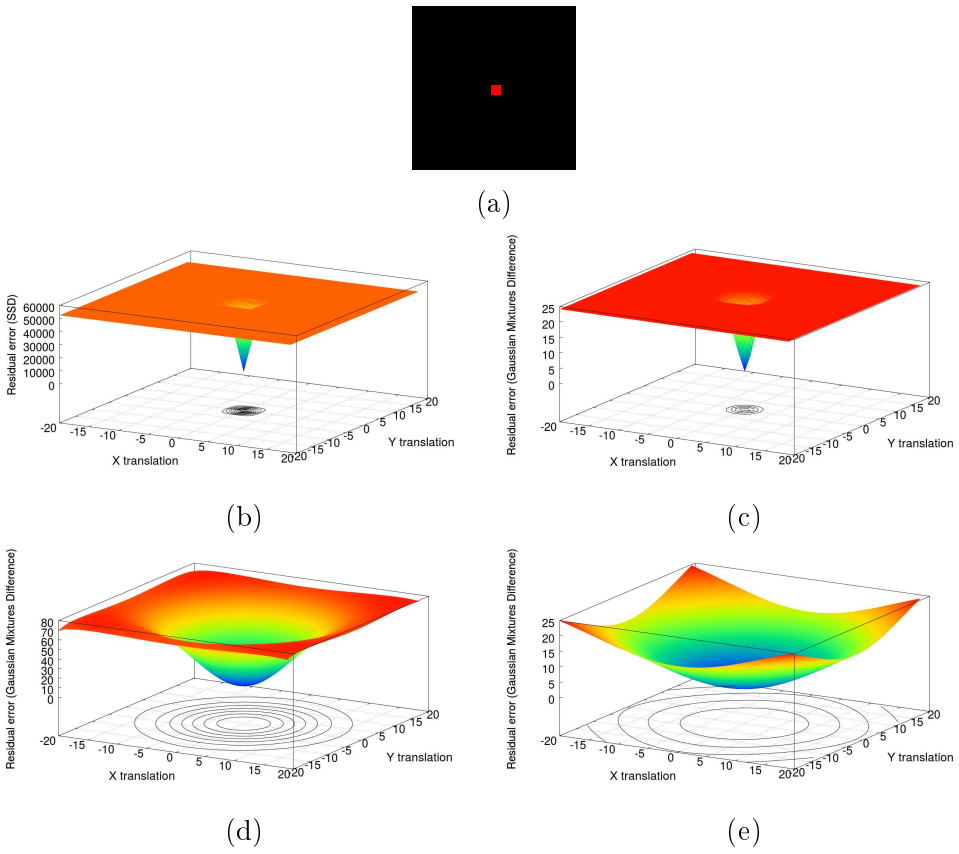


FIGURE 3.4: Comparaisons des fonctions de coût selon 2 ddl : image virtuelle (a), fonction de coût purement photométrique (SSD) (b), fonctions de coût obtenues en représentant les images virtuelles par des mélanges de gaussiennes avec le paramètre d'extension $\lambda_g = \{0.01, 0.05, 0.3\}$ (c-e)

Nous pouvons constater que plus le paramètre d'extension λ_g est grand, plus le domaine de convexité de la fonction de coût obtenue est large. Au contraire, lorsque λ_g est petit, nous pouvons voir que la forme de la fonction de coût obtenue est identique à celle obtenue en utilisant directement l'information photométrique. Cela rejoint le constat effectué précédemment, lorsque le paramètre d'extension est suffisamment petit, le mélange de gaussiennes photométriques d'une image est similaire à l'image elle-même.

Si l'on considère l'exemple de l'image de la figure 3.4a, dans le cas purement photométrique, lorsqu'il n'existe aucun chevauchement entre les projections de la scène à la pose désirée et à la pose courante, la fonction de coût est localement constante. Par conséquent, sa dérivée est nulle et l'erreur photométrique ne peut donc pas être minimisée par les méthodes de type descentes de gradient, Newton,

Levenberg-Marquardt et autres algorithmes d'optimisation non linéaire. D'autres approches stochastiques pourraient éventuellement parvenir à des résultats intéressants mais ce n'est pas l'objet de ce travail. En représentant les images par des mélanges de gaussiennes avec un paramètre d'extension suffisamment grand, un chevauchement entre les deux représentions est alors créé, permettant ainsi la régulation vers 0 de la fonction de coût.

3.3.2 Loi de commande

En considérant la pose de la caméra que l'on souhaite atteindre comme la solution d'un problème d'optimisation non-linéaire, les 6 ddl de la caméra ainsi que le paramètre d'extension des gaussiennes sont déterminés itérativement par une loi de contrôle de type Gauss-Newton :

$$\mathbf{v_g} = \mu \mathbf{L}^{+}_{\mathbf{gm}}(\mathbf{gm}(\mathbf{I}^{*}) - \mathbf{gm}(\mathbf{I}(\mathbf{r}))) \tag{3.7}$$

où $\mathbf{v_g} = (\boldsymbol{\upsilon}\ \boldsymbol{\omega}\ \dot{\lambda}_g)^{\top}$ contient respectivement les vitesses linéaires et angulaires de la caméra et l'incrément du paramètre d'extension des gaussiennes. $\mathbf{L}^{+}_{\mathbf{gm}}$ est la pseudo-inverse de la matrice d'interaction liant les changements entre les mélanges de gaussiennes calculés sur les images $\mathbf{I}(\mathbf{r})$ acquises au cours de l'asservissement par rapport aux déplacements de la caméra. Enfin, le gain μ peut être utilisé pour régler la vitesse de convergence de la caméra.

La matrice d'interaction $\mathbf{L_{gm}}$ est le jacobien du mélange de gaussiennes par rapport à la pose de la caméra. Elle contient l'empilement de tous les vecteurs d'interaction $\mathbf{L}_{gm(\mathbf{u}_g)}$ de chaque valeur du mélange de gaussiennes aux positions $\mathbf{u}_g$:

$$\mathbf{L_{gm}} = \begin{bmatrix} & \vdots & \\ \mathbf{L}_{gm(\mathbf{u}_g)} & & \lambda_{gm(\mathbf{u}_g)} \\ & \vdots & \end{bmatrix}, \tag{3.8}$$

où $\mathbf{I}$ est volontairement omis dans la fonction $gm()$ pour faciliter la lecture. À partir de l'équation (3.5), nous pouvons déduire que $\mathbf{L}_{gm(\mathbf{u}_g)}$ est la somme de toutes les matrices d'interaction associées aux gaussiennes de chaque pixel en $\mathbf{u}_g$:

$$\mathbf{L}_{gm(\mathbf{u}_g)} = \textstyle\sum_{\mathbf{u}} \mathbf{L}_{g(\mathbf{u}_g,\mathbf{u})}, \tag{3.9}$$

où $I_{\mathbf{u}}$ et λ_g sont volontairement omis dans la fonction $g()$ pour faciliter la lecture. La matrice d'interaction $\mathbf{L}_{g(\mathbf{u}_g,\mathbf{u})}$ relie les variations de $g(\mathbf{u}_g, \mathbf{u})$ aux déplacements de la caméra.

Comme pour l'asservissement visuel purement photométrique, la modélisation de la matrice d'interaction $L_{\mathbf{gm}}(\mathbf{x})$ se base sur l'hypothèse que l'intensité I

d'un pixel $\mathbf{u}$ ne varie pas pour un faible déplacement $\delta\mathbf{u}$:

$$I(\mathbf{u}+\delta\mathbf{u}, t+\delta t) = I(\mathbf{u}, t) \tag{3.10}$$

Nous étendons cette relation à la valeur du mélange de gaussiennes $gm(\mathbf{I}(\mathbf{r}), \mathbf{u_g})$ représentant $I(\mathbf{u})$:

$$g(\mathbf{u_g}+\delta\mathbf{u}_g, t+\delta t) = g(\mathbf{u}_g, t), \tag{3.11}$$

où $\mathbf{u}$ est omis dans la fonction $g()$ par soucis de compréhension.

Un développement de Taylor du premier ordre de l'équation (3.11), nous donne :

$$\begin{aligned} g(\mathbf{u_g}, t) &= g(\mathbf{u}_g+\delta\mathbf{u}_g, t+\delta t) \\ &+\frac{\partial g(\mathbf{u}_g+\delta\mathbf{u}_g, t+\delta t)}{\partial \mathbf{u}_g}\frac{\mathrm{d}\mathbf{u}_g}{\mathrm{d}t} \\ &+\frac{\partial g(\mathbf{u}_g+\delta\mathbf{u}_g, t+\delta t)}{\partial t}\frac{\mathrm{d}t}{\mathrm{d}t}, \end{aligned} \tag{3.12}$$

L'égalité de l'équation (3.11), nous permet d'écrire :

$$\frac{\partial g(\mathbf{u}_g+\delta\mathbf{u}_g, t+\delta t)}{\partial \mathbf{u}_g}\frac{\mathrm{d}\mathbf{u}_g}{\mathrm{d}t} + \frac{\partial g(\mathbf{u}_g+\delta\mathbf{u}_g, t+\delta t)}{\partial t} = 0, \tag{3.13}$$

Ce qui peut alors s'écrire sous forme compacte :

$$\nabla g^{\mathsf{T}}\dot{\mathbf{u}}_g + \dot{g} = 0 \Leftrightarrow \dot{g} = -\nabla g^{\mathsf{T}}\dot{\mathbf{u}}_g, \tag{3.14}$$

avec ∇g^{T} le gradient spatial de $g(\mathbf{u}_g+\delta\mathbf{u}_g, t+\delta t)$ et $\dot{g}$ le gradient temporel, qui n'est pas sans rappeler l'équation de la contrainte du flot optique [Horn 1980].

Les variations d'une valeur du mélange de gaussiennes représentant un pixel $\dot{\mathbf{u}}_g$ sont liées aux vitesses de translation et de rotation de la caméra $\mathbf{v} = (\boldsymbol{v}, \boldsymbol{\omega})$ par l'expression :

$$\dot{\mathbf{u}}_g = \mathbf{L}_{\mathbf{u}_g}\mathbf{v} \tag{3.15}$$

En connaissant les paramètres intrinsèques de la caméra (eq. 1.3), $\mathbf{L}_{\mathbf{u}_g}$ peut se décomposer comme étant :

$$\mathbf{L}_{\mathbf{u}_g} = \begin{bmatrix} \alpha_u & 0 \\ 0 & \alpha_v \end{bmatrix}\mathbf{L}_{\mathbf{x}}, \tag{3.16}$$

avec $\mathbf{L}_{\mathbf{x}}$ la matrice d'interaction associée au point exprimé dans le plan image normalisé du modèle de projection de la caméra.

En injectant l'équation (3.15) dans l'équation (3.14), nous obtenons :

$$\dot{g} = -\nabla g^{\mathsf{T}}\mathbf{L}_{\mathbf{u}_g}\mathbf{v}, \tag{3.17}$$

ce qui nous permet d'identifier l'expression de $\mathbf{L}_{g(\mathbf{u}_g,\mathbf{u})}$ (eq. (3.9)) comme étant :

$$\mathbf{L}_{g(\mathbf{u}_g,\mathbf{u})} = -\nabla g^{\top} \mathbf{L}_{\mathbf{u}_g} \tag{3.18}$$

avec $\mathbf{L}_{\mathbf{u}_g}$ connu (eq. (3.16)) et $\nabla g = (\nabla g_u, \nabla g_v)^T$ qui reste à déterminer.

Nous avons vu qu'en asservissement visuel purement photométrique (Section 1.2.2.1), le gradient spatial de l'image est la seule donnée qui résulte d'un traitement d'image. Dans notre cas, l'image est modélisée par un mélange de gaussiennes photométriques dont la formulation est analytiquement connue. Par conséquent, les dérivées de chaque gaussienne peuvent être analytiquement exprimées :

$$\begin{aligned} \nabla g_u &= \frac{\delta g(\mathbf{u}_g, \mathbf{u}, I_{\mathbf{u}}, \lambda_g)}{\delta u_g} \\ &= -\tfrac{(u_g-u)}{(I_{\mathbf{u}}^2 \lambda_g^2)} exp(-\tfrac{(u_g-u)^2+(v_g-v)^2}{2\lambda_g^2 I_{\mathbf{u}}^2}) \end{aligned} \tag{3.19}$$

et

$$\begin{aligned} \nabla g_v &= \frac{\delta g(\mathbf{u}_g, \mathbf{u}, I_{\mathbf{u}}, \lambda_g)}{\delta v_g} \\ &= -\tfrac{(v_g-v)}{(I_{\mathbf{u}}^2 \lambda_g^2)} \exp(-\tfrac{(u_g-u)^2+(v_g-v)^2}{2\lambda_g^2 I_{\mathbf{u}}^2}). \end{aligned} \tag{3.20}$$

Nous avons vu que l'extension des gaussiennes a une influence importante sur le mélange de gaussiennes représentant une image (Figure 3.3), et donc a fortiori, sur le comportement de l'asservissement visuel. Le mélange de gaussiennes de l'image désirée $\mathbf{gm}(\mathbf{I}^*)$ est calculé à la première itération de l'optimisation avec un paramètre d'extension λ_g^* fixé. Concernant les mélanges de gaussiennes des images acquises au cours de l'asservissement $\mathbf{gm}(\mathbf{I}(\mathbf{r}))$, il est possible de faire varier l'extension des gaussiennes en agissant sur le paramètre λ_g. Au même titre que les vitesses à envoyer au robot contrôlant la caméra, λ_g est calculé à chaque itération de l'optimisation. Pour cela, la matrice d'interaction $\mathbf{L}_{\mathbf{gm}}$ (eq 3.8) contient les dérivées de l'équation du modèle de gaussiennes par rapport à λ_g :

$$\begin{aligned} \nabla g_{\lambda_g} &= \frac{\delta g(\mathbf{u}_g, \mathbf{u}, I_{\mathbf{u}}, \lambda_g)}{\delta \lambda_g} \\ &= \frac{(u_g-u)^2+(v_g-v)^2}{I_{\mathbf{u}}^2 \lambda_g^3} \exp(-\frac{(u_g-u)^2+(v_g-v)^2}{2\lambda_g I_{\mathbf{u}}^2}) \end{aligned} \tag{3.21}$$

L'idéal est de choisir un λ_g^* suffisamment petit pour conserver l'information photométrique de l'image désirée $\mathbf{I}^*$, mais assez grand pour garantir un pouvoir d'attraction intéressant aux pixels. Enfin, initialiser l'asservissement avec un paramètre d'extension λ_g élevé permet d'assurer un large chevauchement entre

$\mathbf{gm}(\mathbf{I}^*)$ et $\mathbf{gm}(\mathbf{I}(\mathbf{r}))$ et ainsi élargir le domaine de convergence de la caméra. L'initialisation des paramètres d'extensions des gaussiennes est discutée plus en détail dans les expérimentations.

3.4 Résultats

Cette section présente des résultats d'asservissements visuels basés mélanges de gaussiennes. Que ce soit en simulation ou au cours d'expérimentations réelles, l'approche est évaluée en contrôlant deux, trois ainsi que les six degrés de liberté de la caméra dans différents types de scène.

3.4.1 Simulations

Les expérimentations en simulation sont développées en C++ en utilisant, en partie, la librairie ViSP [Marchand 2005] ainsi que le moteur graphique Ogre3D.

3.4.1.1 Validations pour 2 ddl

La première série d'expérimentations est volontairement menée dans des environnements virtuels simples et en contrôlant uniquement 2 ddl de la caméra afin de valider la méthode.

- Simulation 1 :

Cet exemple a pour but d'illustrer le concept de pouvoir d'attraction attribué aux pixels des images. La scène est minimaliste, elle est composée d'un seul objet plan de couleur unie, fronto-parallèle à la caméra virtuelle.

À la pose de caméra que l'on souhaite atteindre, l'objet de la scène est projeté dans le coin supérieur gauche de l'image virtuelle désirée $\mathbf{I}^*$ (Figure 3.5a). Tandis qu'à la pose de caméra initiale, l'objet de la scène est projeté dans le coin inférieur droit de l'image virtuelle $\mathbf{I}(\mathbf{r}_0)$ (Figure 3.5b) où $\mathbf{r}_0$ représente la pose initiale de la caméra virtuelle, autrement dit, à l'itération 0 de l'optimisation. Avec cette configuration, il est clairement visible qu'il n'existe aucun chevauchement entre la projection de l'objet dans l'image désirée et dans l'image initiale. Considérant la pose optimale de la caméra comme étant la solution d'un problème d'optimisation non-linéaire, différentes méthodes itératives peuvent être employées pour atteindre l'objectif. Ces méthodes consistent à déterminer localement la direction de descente à emprunter dans l'espace de solution, afin de trouver l'itération suivante qui permet à la fonction de coût à minimiser de prendre une valeur inférieure à celle qu'elle a à l'itération courante. C'est pourquoi, en asservissement visuel la convergence vers cette solution dépend directement de la distance entre

la pose initiale de la caméra et la pose désirée. Pour cette expérimentation, il est clair que la fonction de coût purement photométrique est localement constante et ne peut donc pas être minimisée.

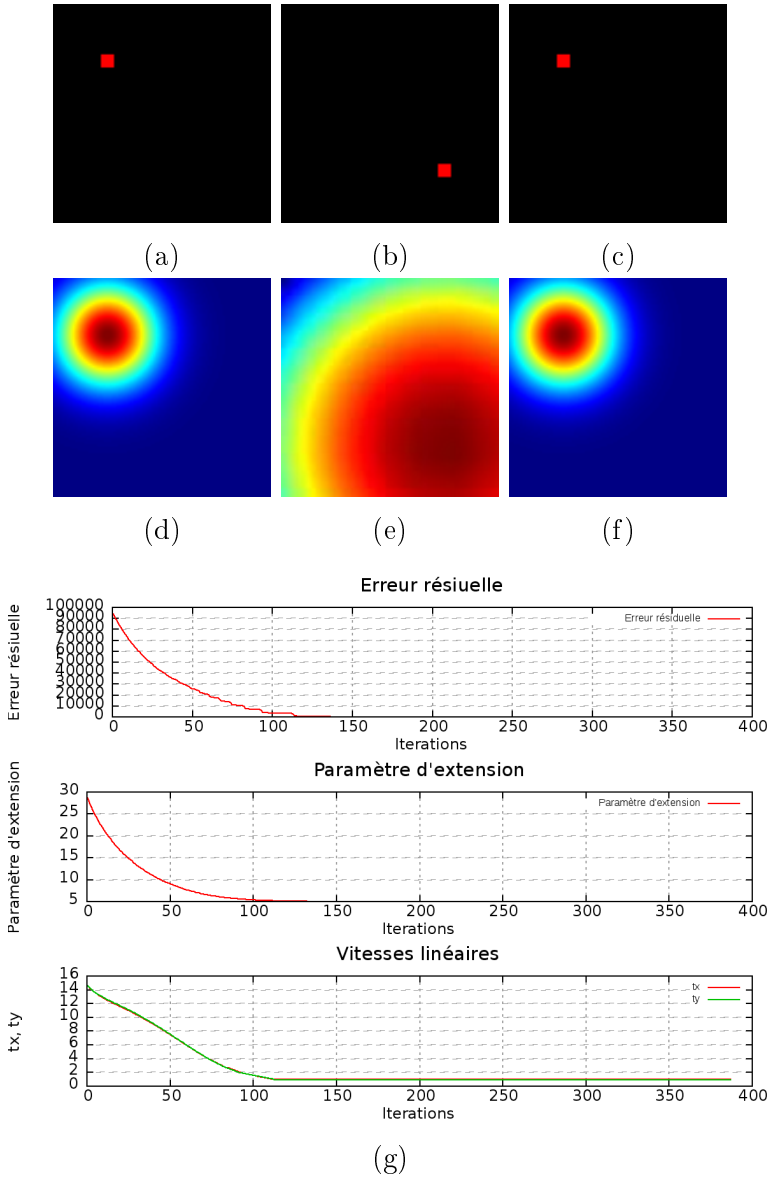


FIGURE 3.5: Simulation 1 pour 2 ddl : image virtuelle désirée (a), initiale (b) et finale (c). Mélange de gaussiennes désiré (d), initial (e) et final (f). Évolution de l'erreur résiduelle, du paramètre d'extension λ_g et des vitesses envoyées à la caméra (g)

Le mélange de gaussiennes de l'image désirée $\mathbf{gm}(\mathbf{I}^*)$ (Figure 3.5d) est calculé en fixant le paramètre d'extension des gaussiennes λ_g^* à 0.3. Le mélange de gaussiennes de l'image initiale $\mathbf{gm}(\mathbf{I}(\mathbf{r}_0))$ (Figure 3.5e) est calculée avec un paramètre

d'extension des gaussiennes λ_g de 5.0. Comme il a été mentionné précédemment, λ_g^* doit être assez faible pour que l'asservissement basé mélanges de gaussiennes ait une précision à convergence proche d'un asservissement visuel photométrique. Quant au λ_g initial, il est élevé afin d'attribuer aux pixels de $\mathbf{I}(\mathbf{r}_0)$ un fort pouvoir d'attraction et ainsi de créer une zone de chevauchement entre les mélanges de gaussiennes de l'image désirée et de l'image initiale pouvant être réduite à l'itération suivante.

Nous pouvons observer que l'erreur entre le mélange de gaussiennes désiré et les mélanges de gaussiennes calculés sur les images acquises tout au long de l'asservissement est correctement minimisée et décroit jusqu'à atteindre zéro (Figure 3.5g). En principe, l'erreur devrait décroître de façon exponentielle, le fait que ce ne soit pas le cas reflète certainement une approximation au niveau de la modélisation de la matrice d'interaction. L'image virtuelle générée à la pose finale de l'optimisation (Figure 3.5c) montre que la caméra a bien convergé vers la pose de la caméra initiale, tout comme le paramètre d'extension final λ_g des gaussiennes (Figure 3.5g et Figure 3.12i).

- Simulation 2 :

Cette fois, la scène est composée de plusieurs objets plans, fronto-parallèles à la caméra virtuelle. Cet exemple a pour but de vérifier le bon fonctionnement de la méthode lorsque des éléments de la scène entrent et sortent du champ de vue de la caméra virtuelle au cours de l'asservissement.

À la pose de la caméra que l'on souhaite atteindre, quatre objets de la scène sont dans le champ de vue de la caméra et sont donc visibles dans l'image désirée (Figure 3.6a). Dans l'image virtuelle initiale (Figure 3.6b), parmi les six objets visibles de la scène, trois ne le sont pas dans l'image désirée. Le mélange de gaussiennes désiré (Figure 3.6d) est calculé avec $\lambda_g = 3.0$ et l'initial avec $\lambda_g^* = 0.1$.

Malgré les informations visuelles qui entrent et sortent du champ de vue de la caméra, l'asservissement visuel basé mélanges de gaussiennes photométriques parvient à déplacer la caméra jusqu'à la pose désirée. L'évolution de l'erreur résiduelle en fonction du temps (Figure 3.6g) montre que la caméra a correctement atteint la pose désirée en minimisant l'erreur entre le mélange de gaussiennes désiré et les mélanges de gaussiennes courants. De la même manière, le paramètre d'extension λ_g a convergé vers le paramètre d'extension désiré λ_g^*. L'image virtuelle obtenue à convergence (Figures 3.6c) ainsi que le mélange de gaussiennes qui lui est associé (Figure 3.6f) illustrent que la pose de la caméra à convergence est très proche de la pose désirée.

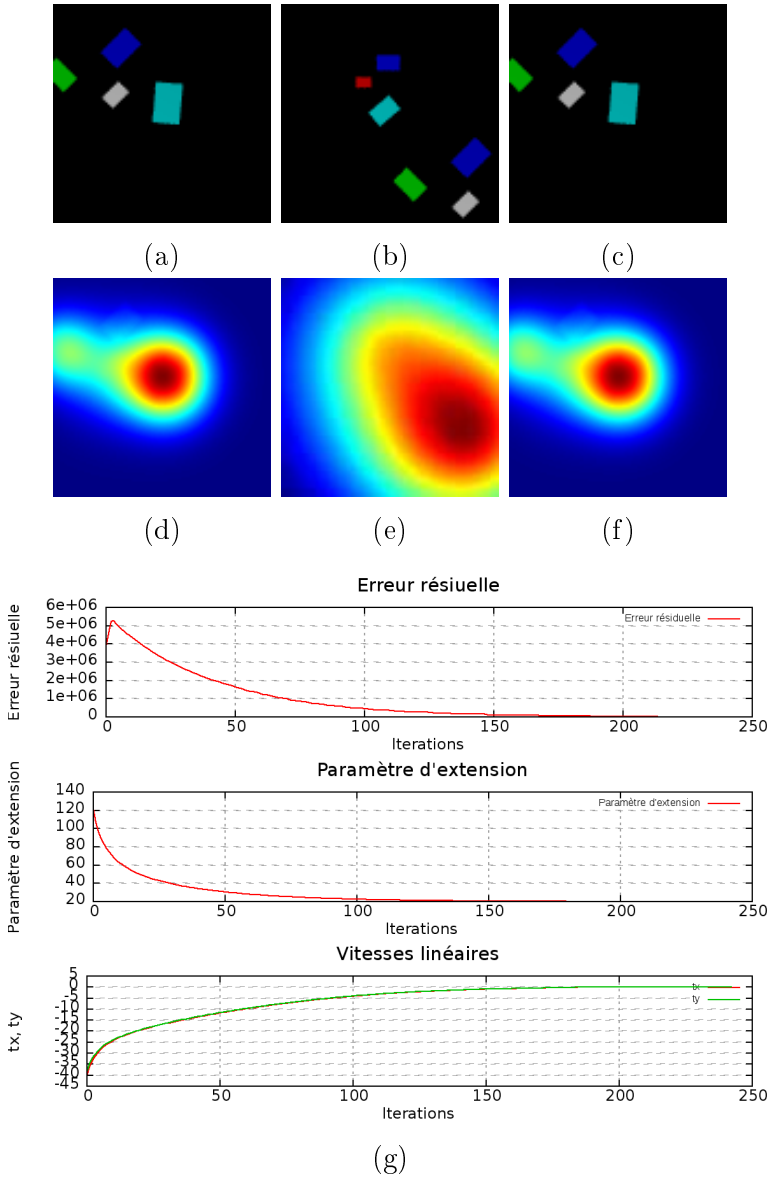


FIGURE 3.6: Simulation 2 pour 2ddl : image virtuelle désirée (a), initiale (b) et finale (c). Mélange de gaussiennes désiré (d), initial (e) et final (f). Évolution de l'erreur résiduelle, du paramètre d'extension λ_g et des vitesses envoyées à la caméra (g)

3.4.1.2 Validations pour 3 ddl

L'asservissement visuel basé mélanges de gaussiennes se montre très intéressant en contrôlant uniquement les translations de la caméra le long des axes ${}^c\vec{X}$ et ${}^c\vec{Y}$ ainsi que les rotations autour de l'axe optique ${}^c\vec{Z}$ de celle-ci.

- Simulation 1 :

Pour cette expérimentation, la scène est composée d'un plan texturé fronto-parallèle à la caméra virtuelle.

La figure 3.7a montre l'image virtuelle générée à partir de la pose désirée. La figure 3.7b montre l'image virtuelle générée à partir de la pose initiale. Il existe une rotation importante autour de l'axe optique de la caméra entre la pose désirée et l'initiale. Plus exactement, l'erreur initiale de positionnement $[\Delta_X, \Delta_Y, \Delta_{R_Z}]$ est de $[-94.55cm, 55.79cm, 69.44°]$. Le mélange de gaussiennes désiré (Figure 3.7d) est calculé avec un paramètre d'extension ${\lambda_g}^* = 0.5$ et l'initial (Figure 3.7e) avec $\lambda_{g_i} = 1.2$.

Malgré la différence initiale importante en position et en orientation, l'asservissement visuel basé mélanges de gaussiennes réussit avec une erreur de positionnement à convergence de $[-0.22cm, -0.10cm, -0.019°]$.

3.4.1.3 Validations pour 6 ddl

- Simulation 1 :

Pour cette première expérimentation à 6 ddl, la scène est composée du même plan texturé que celui utilisé précédemment. Cependant, ce plan n'est plus fronto-parallèle à la caméra virtuelle.

La figure 3.8a montre l'image virtuelle générée à partir de la pose désirée. La figure 3.8b montre l'image virtuelle générée à partir de la pose initiale. Comme on peut le voir sur ces images, il existe des translations et des rotations relativement importantes sur les trois axes séparant la pose initiale de la pose désirée. Plus exactement, l'erreur initiale de positionnement $[\Delta_X, \Delta_Y, \Delta_Z, \Delta_{R_X}, \Delta_{R_Y}, \Delta_{R_Z}]$ est de $[77.65cm, -31.11cm, -54.76cm, 4.59°, -12.05°, -27.56°]$. Le mélange de gaussienne désiré (Figure 3.8d) est calculé avec un paramètre d'extension ${\lambda_g}^* = 0.2$ et l'initial (Figure 3.8e) avec $\lambda_{g_i} = 0.6$.

Comme le montre l'image générée à convergence (Figure 3.8c), l'asservissement visuel basé mélanges de gaussiennes parvient à estimer la pose de la caméra à laquelle l'image désirée (Figure 3.8b) a été générée. L'erreur de positionnement à la fin de l'asservissement est de $[-0.025cm, 0.022cm, -0.17cm, 0.13°, -0.23°, -0.036°]$.

À titre de comparaison, l'asservissement visuel purement photométrique (Section 1.2.2.1) est utilisé à partir de la même configuration de poses de caméra. Avec un tel écart entre la pose désirée et la pose initiale, l'optimisation tombe rapidement dans un minimum local. La figure 3.9a montre l'image de différence à l'état initial, la figure 3.9b montre l'image de différence finale obtenue par asservissement visuel purement photométrique et la figure 3.9c montre l'image de différence finale obtenue par asservissement basé mélanges de gaussiennes pho-

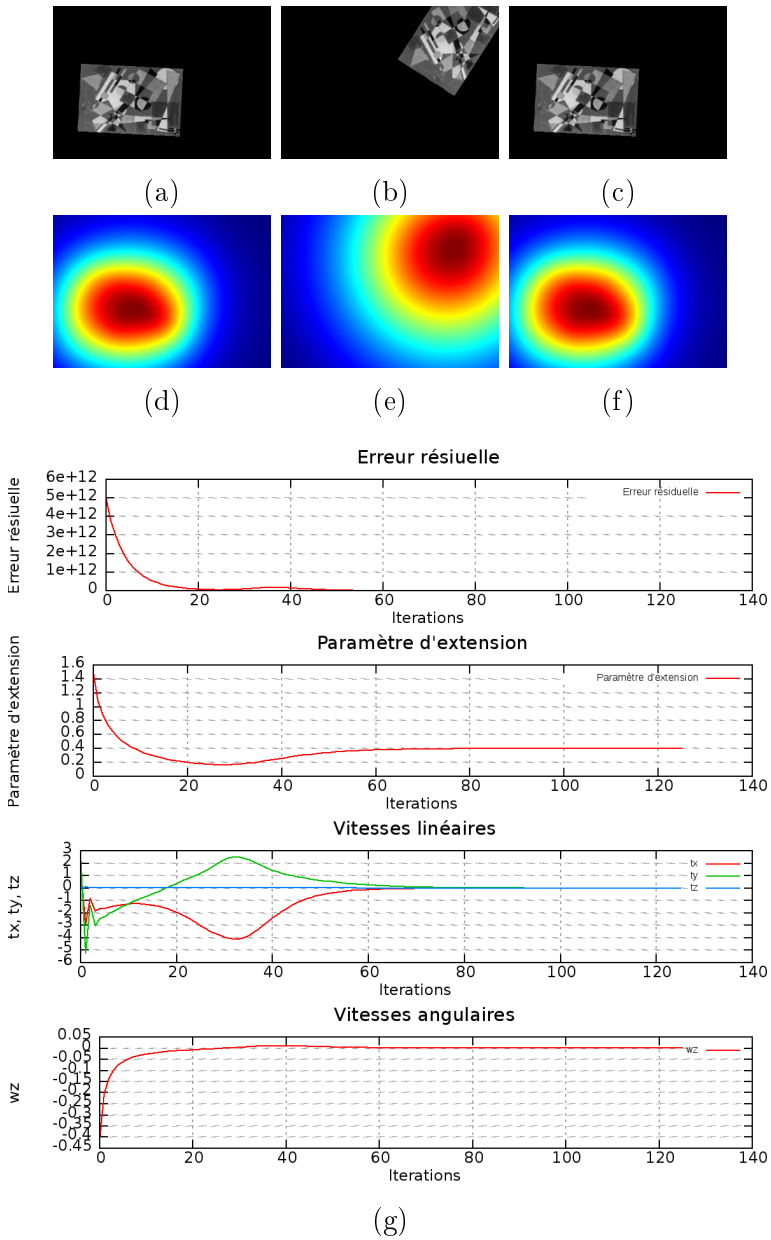


FIGURE 3.7: Simulation 1 pour 3 ddl : image virtuelle désirée (a), initiale (b) et finale (c). Mélange de gaussiennes désiré (d), initial (e) et final (f). Évolution de l'erreur résiduelle, du paramètre d'extension λ_g et des vitesses envoyées à la caméra (g)

tométriques.

- Simulation 2 :

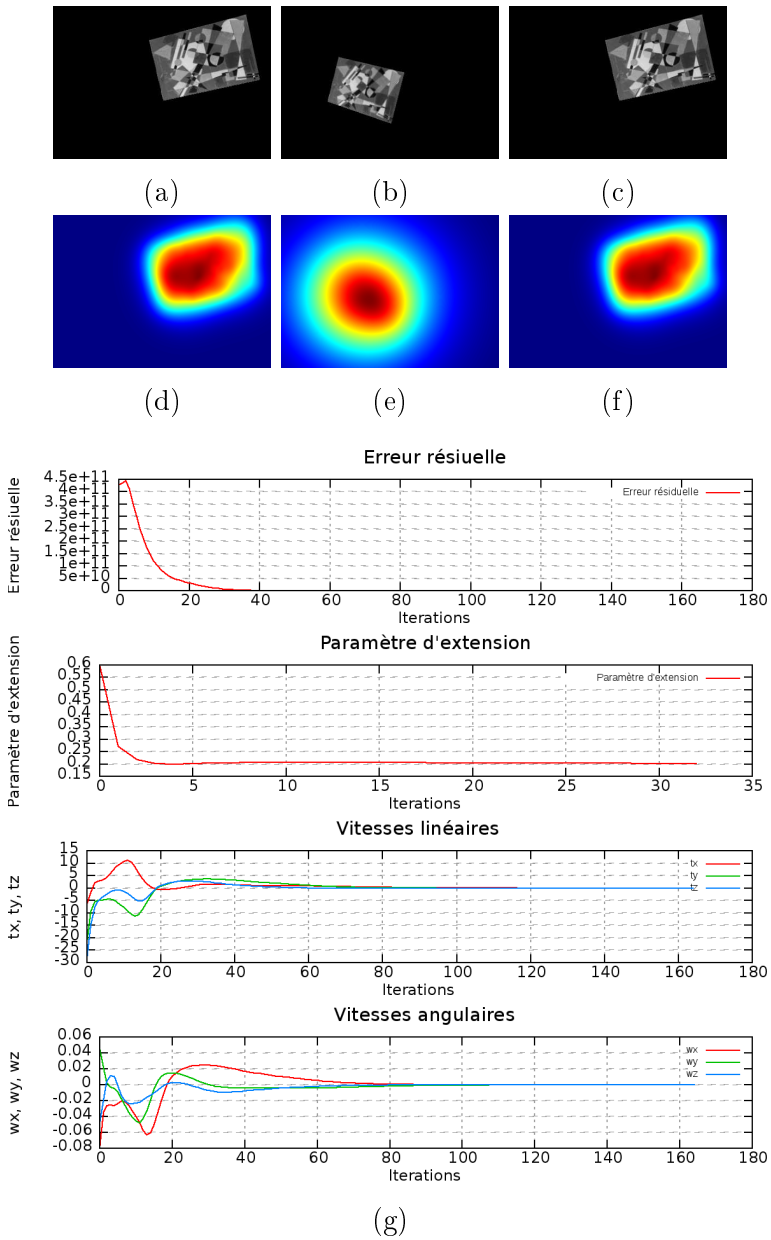
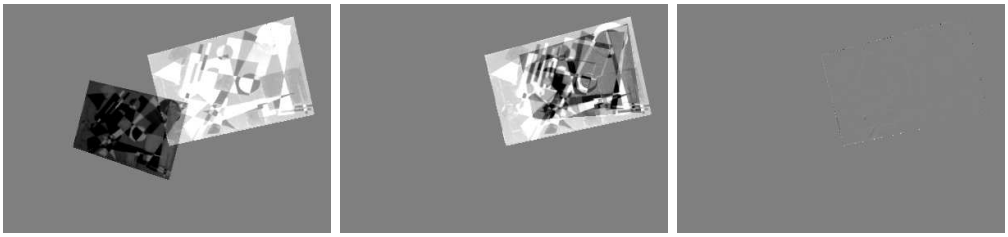


FIGURE 3.8: Simulation 1 pour 6 ddl : image virtuelle désirée (a), initiale (b) et finale (c). Mélange de gaussiennes désiré (d), initial (e) et final (f). Évolution de l'erreur résiduelle, du paramètre d'extension λ_g et des vitesses envoyées à la caméra (g)

Cet exemple se distingue de la première simulation par la nature de la scène virtuelle utilisée qui est en 3D.

La figure 3.10a montre l'image virtuelle générée à partir de la pose désirée. La figure 3.10b montre l'image virtuelle générée à partir de la pose initiale.



(a) (b) (c)

FIGURE 3.9: Simulation 1 pour 6 ddl : image de différence initiale (a), finale par asservissement purement photométrique (b) et finale par asservissement basé mélanges de gaussiennes (c)

Comme on peut le voir sur ces images, il existe des translations et des rotations relativement importantes sur les trois axes séparant la pose initiale de la pose désirée. L'erreur initiale de positionnement $[\Delta_X, \Delta_Y, \Delta_Z, \Delta_{R_X}, \Delta_{R_Y}, \Delta_{R_Z}]$ est de $[-127.33cm, -87.64cm, 67.55cm, -22.52°, -15.20°, -5.46°]$. Le mélange de gaussiennes désiré (Figure 3.10e) est calculé avec un paramètre d'extension ${\lambda_g}^* = 0.05$ et l'initial (Figure 3.10f) avec $\lambda_{g_i} = 0.8$.

Comme le montre l'image générée à convergence (Figure 3.10c), l'asservissement visuel basé mélanges de gaussiennes parvient à estimer la pose de caméra à laquelle l'image désirée (Figure 3.10b) a été générée. L'erreur de positionnement à la fin de l'asservissement est de $[-2.80cm, -0.60cm, 1.32cm, -0.65°, -0.13°, 0.09°]$. La caméra a convergé vers la pose désirée, cependant la pose finale n'est pas parfaite (Figure 3.10h). Plusieurs raisons peuvent expliquer cette légère imprécision à convergence. Tout d'abord, le paramètre d'extension désiré choisi est peut être trop grand, ce qui rend le mélange de gaussiennes désiré moins discriminant que l'image désirée elle même. Ou alors, l'asservissement a peut être été arrêté trop tôt. En effet, les vitesses envoyées à la caméra lorsque celle-ci est très proche de la solution optimale sont très faibles, par conséquent, atteindre la pose désirée peut prendre un grand nombre d'itérations. Dans les deux cas, il est possible de basculer sur un asservissement purement photométrique en partant de la pose finale obtenue par asservissement basé mélanges de gaussiennes.

- Simulation 2 (bis) :

Jusqu'à présent, dans toutes les simulations présentées, la matrice d'interaction $\mathbf{L_{gm}}$ (eq. 3.8) a été calculée à chaque itération de l'optimisation en utilisant les "vraies" profondeurs de la scène virtuelle, c'est-à-dire les informations retournées par le Z-Buffer du moteur graphique. Ce dernier exemple est identique à l'expérimentation précédente à l'exception que les profondeurs de la scène sont considérées comme fixes et similaires en tout point de l'image virtuelle $\mathbf{I}(\mathbf{r})$. La

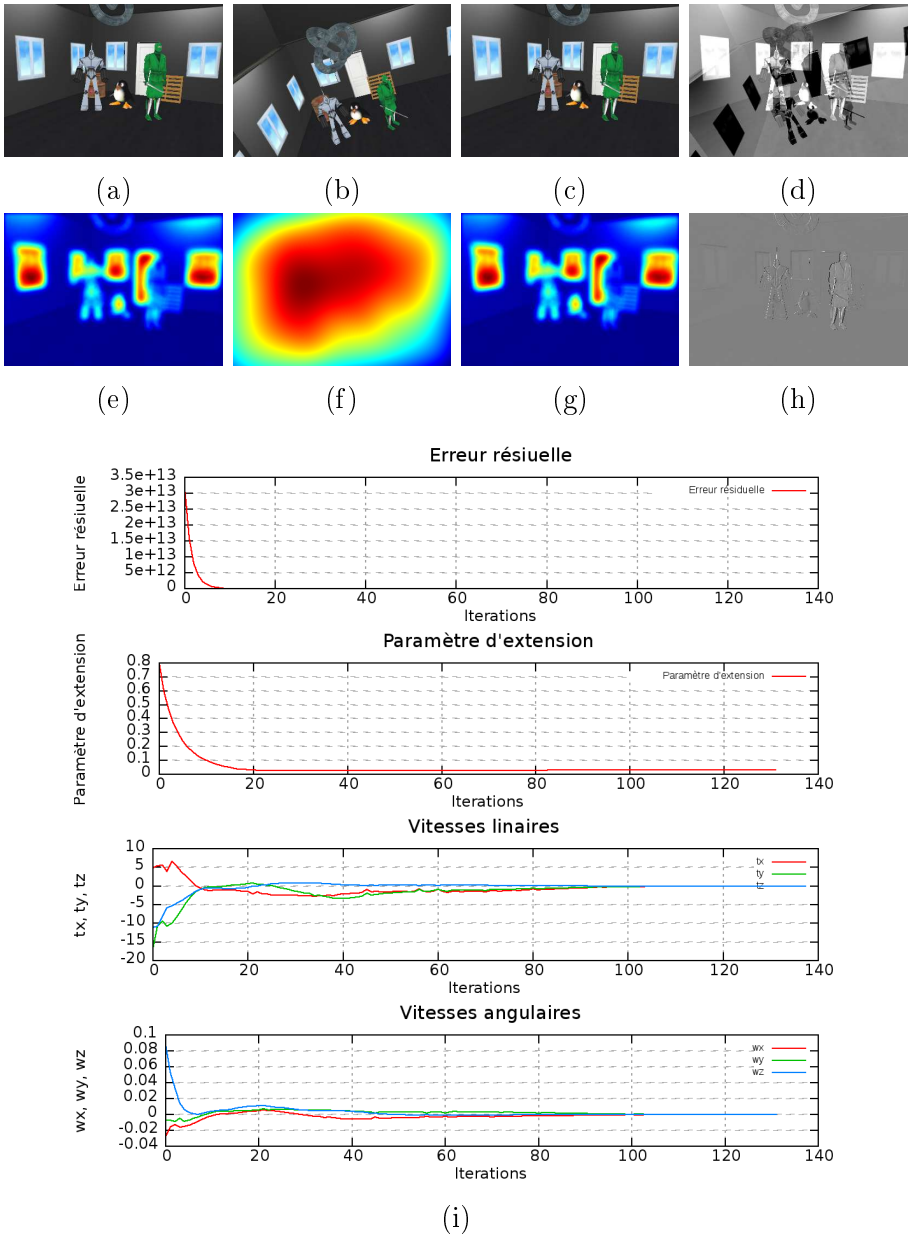


FIGURE 3.10: Simulation 2 pour 6 ddl : image virtuelle désirée (a), initiale (b) et finale (c). Mélange de gaussiennes désiré (d), initial (e) et final (f). Évolution de l'erreur résiduelle, du paramètre d'extension λ_g et des vitesses envoyées à la caméra (g).

profondeur choisie pour toute l'image virtuelle est de trois mètres (distance entre la caméra et la scène au centre de l'image désirée).

Les images désirée et initiale ainsi que le paramètre d'extension initial et désiré des mélanges de gaussiennes sont identiques à ceux de l'expérimentation pré-

cédente (Figures 3.10). Dans ces conditions, le comportement de l'asservissement n'est plus le même et la caméra diverge de la pose attendue. Plusieurs valeurs de profondeur ont été utilisées (un mètre, cinq mètres, dix mètres ...), pour chacune d'elles les déplacements de la caméra pendant l'asservissement sont différents mais ne permettent jamais d'atteindre la solution désirée. Les profondeurs de la scène ayant une grande influence sur le bon fonctionnement de l'asservissement, nous n'utilisons dans les expérimentations réelles qu'une scène plane afin d'avoir une estimation relativement précise des profondeurs à la pose désirée.

3.4.2 Application sur un robot manipulateur

Les expérimentations en environnement réel sont réalisées sur un robot industriel 6 axes (Staubli Tx60) équipé d'une caméra perspective montée sur son effecteur (Figure 3.11). La configuration du système est donc dite "eye-in-hand".

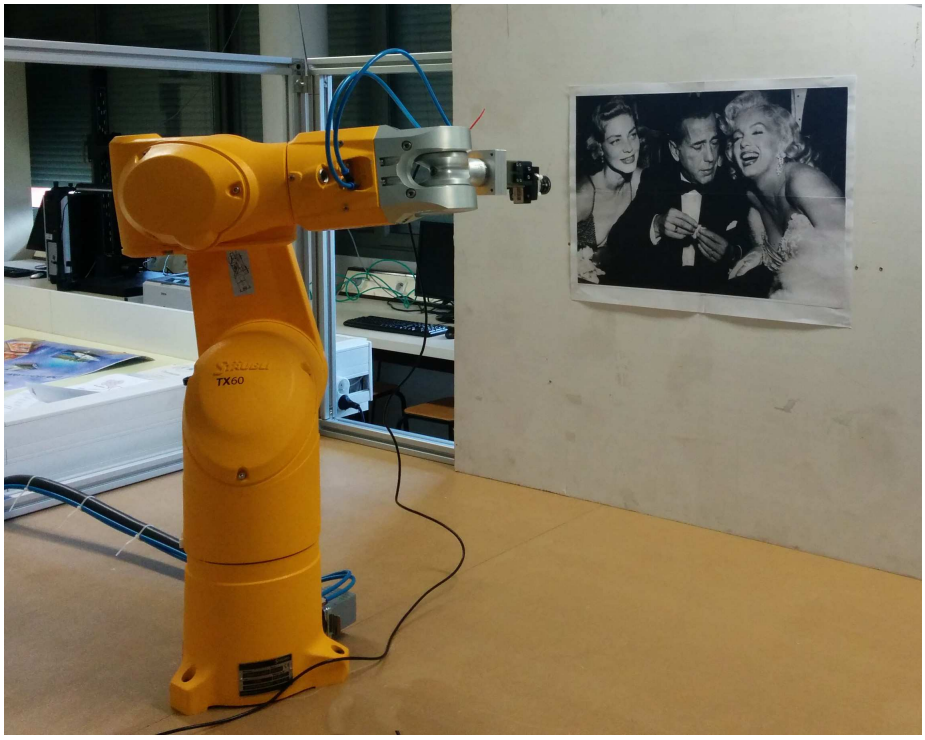


FIGURE 3.11: Environnement dans lequel sont menées les experimentations réelles : robot industriel 6 axes (Staubli Tx60) équipé d'une caméra perspective montée sur son effecteur

Comme en simulation, un premier jeu d'expérimentations a pour objectif de valider le concept et de montrer la réussite de l'asservissement visuel basé mélanges de gaussiennes photométriques en contrôlant trois degrés de liberté de la caméra. Ces expérimentations permettent de mettre en évidence plus facilement la capacité de convergence de l'asservissement pour de grandes erreurs de positionnement initial. Dans un second temps, des expérimentations sur les 6 degrés de liberté sont réalisées. Différentes scènes ont été utilisées afin de démontrer que l'asservissement visuel basé mélanges de gaussiennes photométriques fonctionne indépendamment du contenu des images acquises par la caméra.

Pour toutes les expérimentations, la matrice d'interaction est calculée à chaque itération de l'asservissement à partir du mélange de gaussiennes courant. Nous avons vu en simulation que les profondeurs de la scène intervenant dans le calcul de la matrice d'interaction $\mathbf{L_x}$ (eq. 3.16) ont une influence sur le comportement

de l'asservissement. La profondeur de la scène est, ici, un paramètre inconnu. La scène utilisée dans les expérimentations réelles étant plane, une estimation grossière des profondeurs en position désirée est utilisée pour tous les pixels des images acquises au cours des déplacements de la caméra.

3.4.2.1 Implémentation

Le calcul du mélange de gaussiennes d'une image est une opération chronophage. Pour permettre d'utiliser l'approche dans des expérimentations réelles, les images de taille (1600×1280) acquises par la caméra sont réduites à des images de taille (80×64). Pour accélérer encore l'obtention des mélanges de gaussiennes des images, leur calcul est parallélisé et exécuté sur le processeur graphique (GPU). À titre de comparaison, le calcul du mélange de gaussiennes d'une image de taille (80×64) prend plus de 1200 ms avec un algorithme séquentiel développé en C++. Avec l'algorithme parallélisé, le calcul de ce mélange de gaussiennes ne demande plus qu'environ 100ms sur une machine dotée d'un processeur Intel Core I7 cadencé à 2.3GHz, avec 4Go de RAM et une carte graphique NVIDIA GeForce GT 630M.

Cela reste relativement important mais rend les mélanges de gaussiennes utilisables dans nos expérimentations réelles.

Toutes les étapes du processus de l'asservissement visuel (calcul des gradients, de la matrice d'interaction, du vecteur d'erreur et de la mise à jour de la pose) sont également parallélisées et sont calculées sur le GPU. Les seules informations qui transitent entre le CPU et le GPU sont l'image désirée (transférée une seule fois à l'initialisation) puis à chaque itération, l'image courante et les vitesses.

3.4.2.2 Expérimentations réelles

Expérimentation 1

Pour la première expérimentation, nous essayons de nous placer dans une configuration proche de celles des premières simulations. Pour commencer, nous contrôlons 3 ddl, les translations de la caméra le long des axes ${}^c\vec{X}$ et ${}^c\vec{Y}$ ainsi que les rotations autour de l'axe optique ${}^c\vec{Z}$ de celle-ci. La scène est composée d'une forme géométrique plane non texturée.

La figure 3.12a montre l'image acquise par la caméra lorsque le robot est en position désirée et la figure 3.12b montre l'image acquise par la caméra lorsque le robot est en position initiale. Le mélange de gaussiennes désiré (Figure 3.12f) est calculé avec un paramètre d'extension ${\lambda_g}^* = 0.1$ et l'initial (Figure 3.12g) avec $\lambda_{g_i} = 1.2$. Le déplacement entre la pose désirée et la pose initiale est de $13.3cm$ en translation avec une rotation de 28.41° autour de l'axe optique de la caméra.

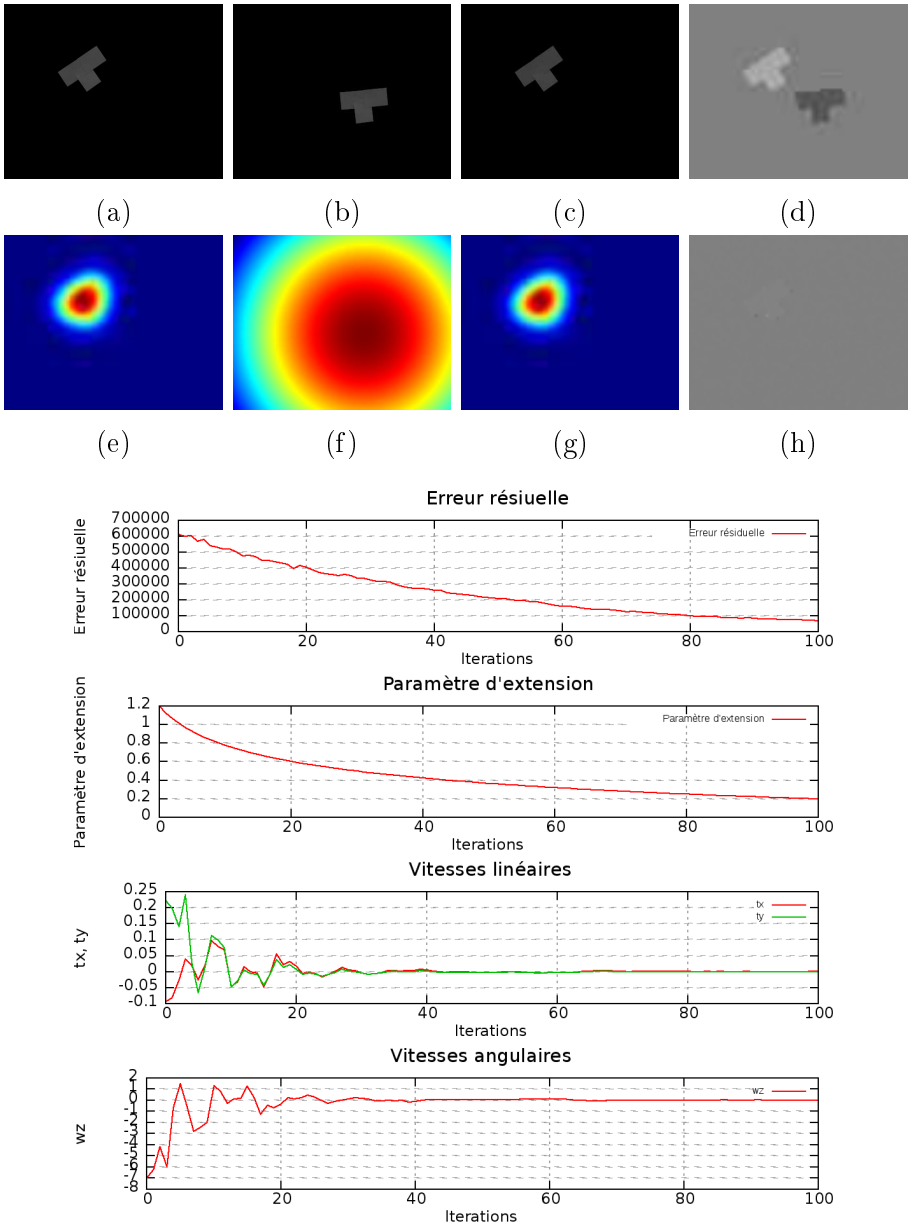


FIGURE 3.12: Expérimentation 1 pour 3 ddl : image désirée (a), initiale (b) et finale (c). Mélange de gaussiennes désiré (d), initial (e) et final (f). Image de différence initiale (d) et finale (h). Évolution de l'erreur résiduelle, du paramètre d'extension λ_g et des vitesses envoyées au robot (i).

Même si les positions de la scène dans l'image initiale et dans l'image désirée sont éloignées, l'asservissement visuel basé mélange de gaussiennes photométriques parvient à faire converger la caméra vers la pose désirée. Plus précisément,

la pose optimale se trouve à moins de $0.07cm$ de la pose désirée et à $0.43°$ de rotation autour de l'axe optique.

Expérimentation 2

Dans cette deuxième expérimentation, nous contrôlons toujours les mêmes 3 ddl mais cette fois la scène est constituée d'une image texturée. L'image est fréquemment utilisée en asservissement visuel [Collewet 2008].

La figure 3.13a montre l'image acquise par la caméra lorsque le robot est en position désirée. La figure 3.13b montre l'image acquise par la caméra lorsque le robot est en position initiale. Le mélange de gaussienne désiré (Figure 3.13e) est calculé avec un paramètre d'extension $\lambda_g{}^* = 0.25$ et l'initial (Figure 3.13f) avec $\lambda_{g_i} = 1.0$. Le déplacement entre la pose désirée et la pose initiale est de $16.08cm$ en translation avec une rotation de $45.87°$ autour de l'axe optique. L'erreur finale, à convergence, est de moins d'$1mm$ et de $0.73°$ en rotation.

Expérimentation 3

Cette dernière expérimentation est réalisée en utilisant la même scène mais en contrôlant les 6 ddl de la caméra.

La figure 3.14a montre l'image acquise par la caméra lorsque le robot est en position désirée. La figure 3.14b montre l'image acquise par la caméra lorsque le robot est en position initiale. Le mélange de gaussienne désiré (Figure 3.14e) est calculé avec un paramètre d'extension $\lambda_g{}^* = 0.1$ et l'initial (Figure 3.14f) avec $\lambda_{g_i} = 0.8$. L'erreur de positionnement initial $(t_x, t_y, t_z, R_x, R_y, R_z)$ est de $(8.22cm, 5.75cm, 5.1cm, 35.17°, 8.50°, 5.75°)$.

L'erreur de positionnement à convergence est $(0.129mm, 1.01mm, 4.28mm, 0.60°, 0.77°, 0.35°)$.

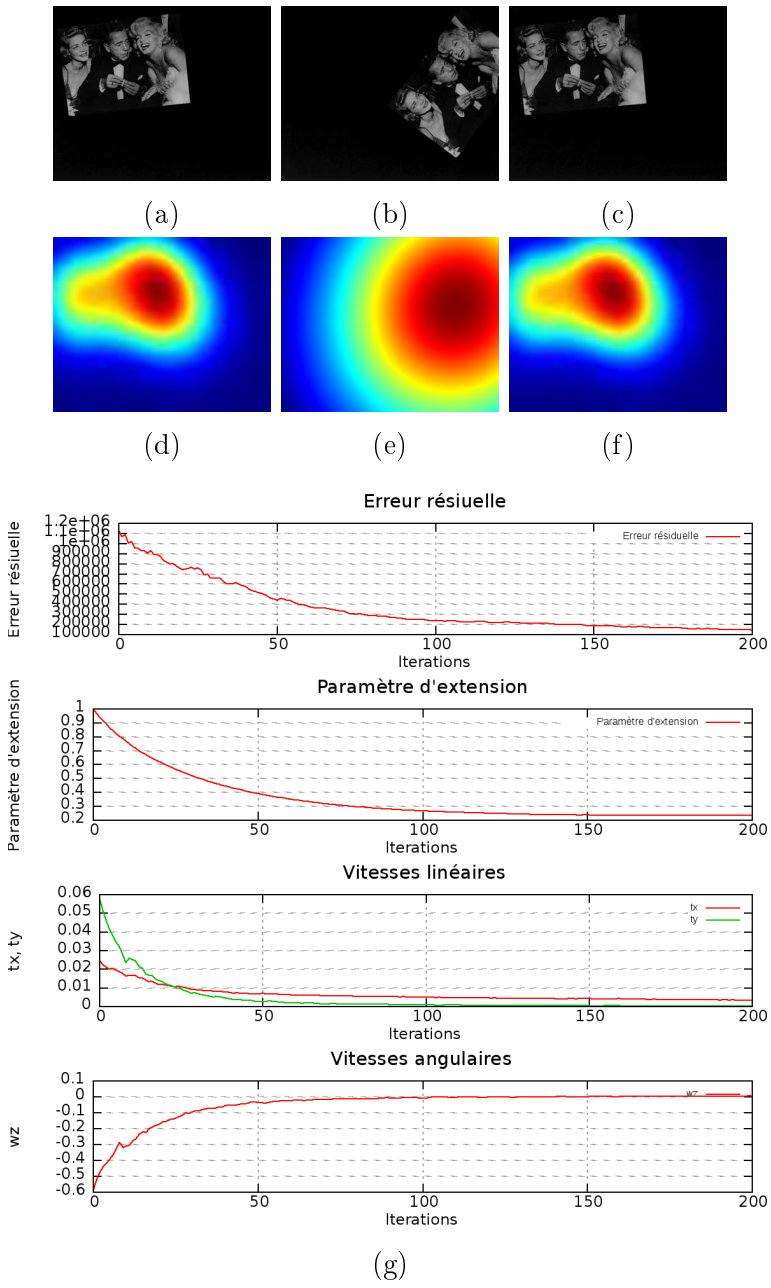


FIGURE 3.13: Expérimentation 2 pour 3 ddl : image désirée (a), initiale (b) et finale (c). Mélange de gaussiennes désiré (d), initial (e) et final (f). Image de différence initiale (d) et finale (h). Évolution de l'erreur résiduelle, du paramètre d'extension λ_g et des vitesses envoyées au robot (i).

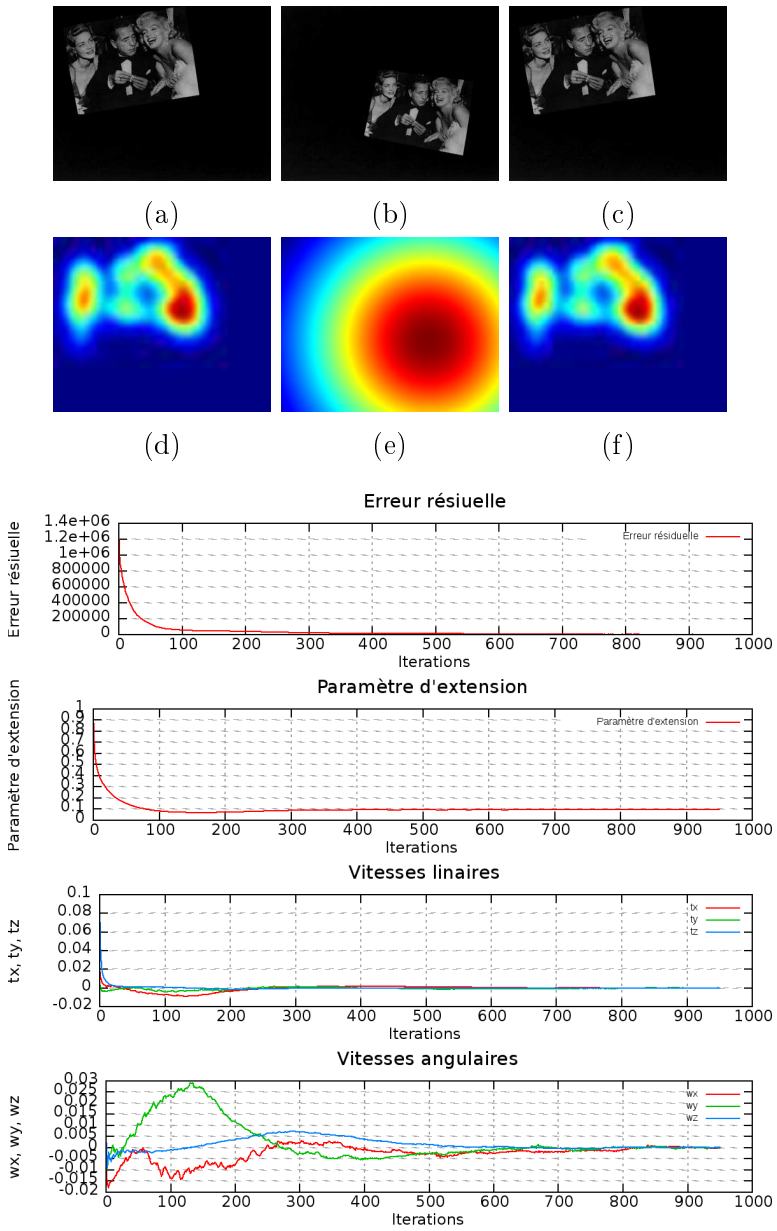


FIGURE 3.14: Expérimentation pour 6 ddl : image désirée (a), initiale (b) et finale (c). Mélange de gaussiennes désiré (d), initial (e) et final (f). Image de différence initiale (d) et finale (h). Évolution de l'erreur résiduelle, du paramètre d'extension λ_g et des vitesses envoyées au robot (i)

3.5 Conclusion

Les mélanges de gaussiennes photométriques ont été introduits en tant que nouvelle caractéristique visuelle dense pour l'asservissement visuel. Les résultats des expérimentations de calcul de pose du deuxième chapitre, et en particulier, les problèmes rencontrés liés au domaine de convergence limité des asservissements visuels photométriques, ont encouragé ces recherches.

Nous avons proposé de représenter un pixel par une fonction gaussienne 2D afin d'attribuer à chaque pixel de l'image un pouvoir d'attraction. La combinaison de toutes les gaussiennes permet de modéliser l'image complète sous la forme d'un mélange de gaussiennes. Au-delà du concept du pouvoir d'attraction, l'intérêt du mélange de gaussiennes est d'être paramétrable. En effet, faire varier l'envergure des gaussiennes permet de passer d'une représentation globale de l'image à un mélange dont les valeurs sont proches des intensités de l'image modélisée. En plus de la commande de la caméra pendant l'asservissement, nous proposons d'optimiser également l'extension des gaussiennes du mélange représentant l'image courante. En choisissant judicieusement l'extension des gaussiennes du mélange de l'image désirée et celle du mélange de l'image initiale, il est alors possible de commencer l'asservissement avec des pixels bénéficiant d'un fort pouvoir d'attraction et de le finir avec des mélanges représentant les images sous une forme quasiment photométrique.

La matrice d'interaction a été développée pour cette nouvelle modélisation des images. Pour l'asservissement visuel purement photométrique, le gradient spatial de l'image est la seule donnée qui résulte d'un traitement d'image. Ici, nous connaissons une formulation analytique du mélange de gaussiennes représentant l'image, la matrice d'interaction est donc développée en conséquence.

Une simple loi de commande du premier ordre est suffisante pour réaliser le contrôle des degrés de liberté de la caméra et pour estimer l'évolution de l'extension des gaussiennes. De nombreuses simulations et expérimentations réelles ont été menées afin de valider cette nouvelle approche. Que ce soit pour deux, pour trois ou pour six degrés de liberté, et dans différents types d'environnement, les mélanges de gaussiennes photométriques permettent d'élargir significativement le domaine de convergence de l'asservissement tout en conservant la grande précision à convergence des asservissements photométriques.

Chapitre 4

Asservissement visuel virtuel basé mélanges de gaussiennes photométriques

Sommaire

4.1 Introduction

Ce chapitre a pour objectif d'étendre l'utilisation des mélanges de gaussiennes photométriques comme caractéristiques visuelles aux asservissements visuels virtuels. Comme dans le chapitre 2, nos travaux sont toujours réalisés dans le cadre du programme de recherche E-Cathédrale. C'est pourquoi, dans nos expérimentations, nous utilisons comme représentation virtuelle de l'environnement un nuage de points 3D acquis à partir de scanners laser. Cependant, ce type de modèle n'est pas une limitation de la méthode proposée. Dès l'instant où d'autres types de modèles 3D possèdent une information photométrique de l'environnement qu'ils représentent, ces modèles peuvent également être utilisés.

Les études précédemment menées ainsi que les résultats en asservissement visuel basé mélanges de gaussiennes photométriques (Chapitre 3) nous encouragent

à penser que les mélanges de gaussiennes pourraient être des caractéristiques visuelles denses très intéressantes, compte-tenu des qualités et des défauts inhérents aux modèles 3D issus de prélèvements par scanners laser.

Plusieurs problèmes ont été soulevés durant les différentes applications réalisées à partir d'asservissement visuel virtuel photométrique basé nuages de points colorés (Section 2.5). Ces difficultés sont, pour la plupart, étroitement liées au domaine de convergence limité de l'asservissement visuel virtuel photométrique et également à la mauvaise qualité visuelle des images générées à partir des modèles 3D composés de plusieurs nuages de points provenant de différentes stations. La représentation des images par des mélanges de gaussiennes photométriques est une solution intéressante à ces problèmes. En effet, nous avons vu dans le chapitre précédent que les mélanges de gaussiennes utilisés en tant que caractéristiques visuelles denses permettent d'accroître significativement le domaine de convergence de l'asservissement visuel. Qui plus est, nous avons également vu que le mélange de gaussiennes d'une image calculé avec un grand paramètre d'extension est une représentation globale du contenu de l'image. Par conséquent, le mélange de gaussiennes d'une image numérique et le mélange de gaussiennes d'une image virtuelle pourraient être proches malgré la mauvaise qualité photométrique de l'image virtuelle.

4.2 Modèles de gaussienne photométrique

Dans le chapitre précédent, nous avons défini les paramètres de la fonction gaussienne représentant un pixel d'une image (eq. 3.4). Cependant, d'autres jeux de paramètres décrivant la fonction gaussienne de chaque pixel sont également envisageables.

Nous définissons un modèle de gaussienne par les trois paramètres $\{A, \mathbf{u_0}, \boldsymbol{\sigma}\}$ qui représentent respectivement l'amplitude, le centre et l'envergure de la gaussienne. Nous discutons et comparons ici trois modèles, le premier étant un rappel du modèle utilisé dans le chapitre précédent.

4.2.1 Introduction des modèles

4.2.1.1 Modèle 1 : Intensité/Envergure

Dans ce modèle, la gaussienne représentant un pixel est centrée sur la position de ce pixel dans l'image. Par conséquent, pour un pixel situé aux coordonnées $\mathbf{u} = (u, v)$ le paramètre $\mathbf{u}_g$ de l'équation 3.3 est égal à $\mathbf{u}$.

Toujours pour un pixel aux coordonnées $\mathbf{u} = (u, v)$, l'envergure de la gaussienne $\boldsymbol{\sigma}$ le long des axes $\vec{u}$ et $\vec{v}$ sont choisis comme égaux et proportionnels à

l'intensité $I_{\mathbf{u}}$ du pixel $\mathbf{u}$. Ce choix est motivé par l'envie de conserver la distinction entre les différentes intensités contenues dans l'image. D'un point de vue pratique, durant l'asservissement visuel, ce choix a pour but de créer une "attraction" entre les pixels d'intensité similaire, a fortiori, entre les gaussiennes d'envergure similaire.

L'amplitude A de toutes les gaussiennes est fixée à 1. Pour résumer, chaque gaussienne est centrée sur le pixel qu'elle représente, avec une amplitude égale à 1 et une envergure différente en fonction de l'intensité du pixel représenté. La fonction gaussienne photométrique de ce modèle est alors :

$$g_1(\mathbf{u}_g, \mathbf{u}) = \exp\left(-\left(\frac{(u_g - u)^2 + (v_g - v)^2}{2\lambda_g^2 I^2(\mathbf{u})}\right)\right) \tag{4.1}$$

où λ_g est toujours le paramètre d'extension de la gaussienne.

La figure 4.1 montre des mélanges de gaussiennes obtenus avec ce modèle pour différentes valeurs d'extension. Par souci de compréhension, ces mélanges

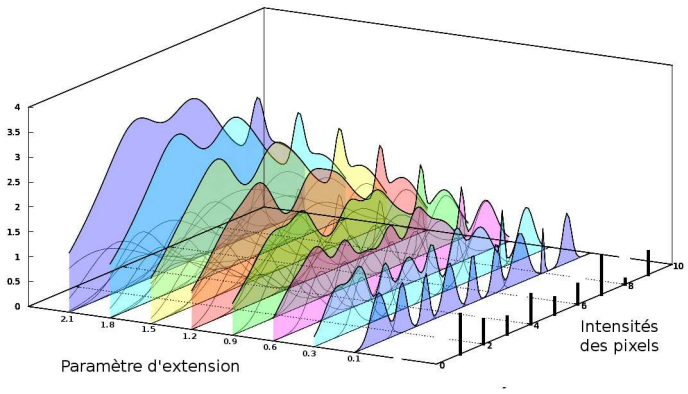


FIGURE 4.1: Modèle de gaussienne Intensité/Envergure

de gaussiennes sont calculés à partir d'un signal 1D et non d'une image 2D complète, mais le principe reste le même.

4.2.1.2 Modèle 2 : Intensité/Envergure (Normalisé)

En modifiant l'amplitude A du modèle précédent, il est possible de normaliser les fonctions gaussiennes représentant chaque pixel de l'image. Avec $A = \frac{1}{\sqrt{2\pi}\lambda_g^2 I^2(\mathbf{u})}$, la valeur de l'amplitude d'une gaussienne dépend de son envergure, ainsi l'aire sous la gaussienne est alors égale à 1. La fonction gaussienne

photométrique de ce modèle est alors :

$$g_2(\mathbf{u}_g, \mathbf{u}) = \frac{1}{\sqrt{2\pi}\lambda_g^2 I^2(\mathbf{u})} \exp\left(-\left(\frac{(u_g - u)^2 + (v_g - v)^2}{2\lambda_g^2 I^2(\mathbf{u})}\right)\right) \tag{4.2}$$

La figure 4.2 montre des mélanges de gaussiennes obtenus avec ce modèle pour différentes valeurs d'extension. Le signal 1D est identique à celui utilisé pour

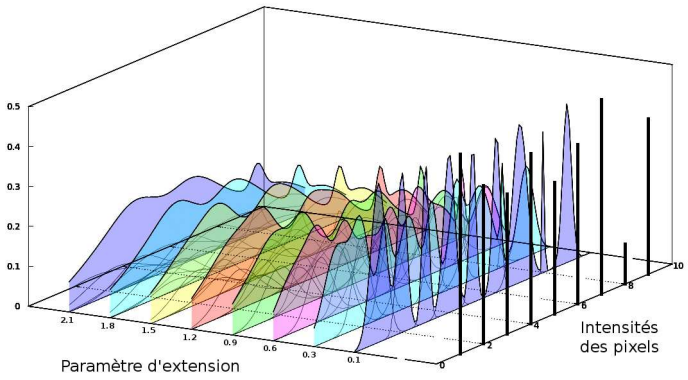


FIGURE 4.2: Modèle de gaussienne Intensité/Envergure (Normalisé)

calculer les mélanges de gaussiennes avec le modèle précédent (Figure 4.1).

4.2.1.3 Modèle 3 : Intensité/Amplitude

Comme pour les modèles précédents, la gaussienne représentant un pixel est centrée sur la position de ce pixel dans l'image, donc $\mathbf{u}_g = \mathbf{u}$.

Tous les pixels d'une image sont représentés par des gaussiennes ayant une même envergure. L'envergure le long des axes $\vec{u}$ et $\vec{v}$ sont égaux $\sigma_u = \sigma_v = \sigma$. Cette envergure est toujours pondérée par le paramètre d'extension λ_g.

Cette fois, pour conserver la distinction entre les différentes intensités contenues dans l'image, c'est l'amplitude de la gaussienne représentant un pixel $\mathbf{u}$ qui est égale à l'intensité $I(\mathbf{u})$ de ce pixel. La fonction gaussienne photométrique de ce modèle s'écrit alors :

$$g_3(\mathbf{u}_g, \mathbf{u}) = I(\mathbf{u}) \exp\left(-\left(\frac{(u_g - u)^2 + (v_g - v)^2}{2\lambda_g^2 \sigma^2}\right)\right) \tag{4.3}$$

La figure 4.3 montre des mélanges de gaussiennes obtenus avec ce modèle pour différentes valeurs d'extension. Le signal en entrée est identique à celui utilisé pour calculer les mélanges de gaussiennes avec les modèles précédents (Figures 4.1 et 4.2).

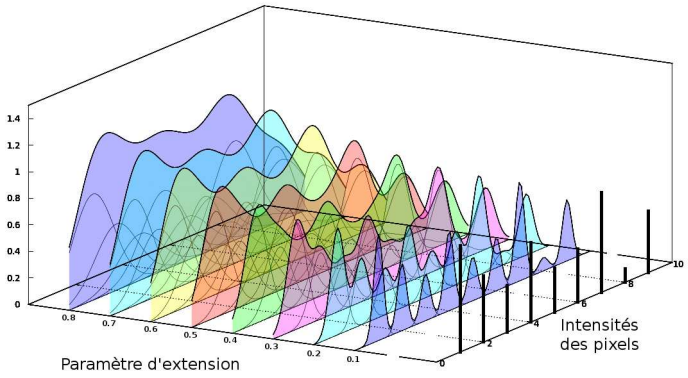


FIGURE 4.3: Modèle de gaussienne Intensité/Amplitude

4.2.2 Influence du modèle sur les mélanges de gaussiennes

Un mélange de gaussiennes étant la combinaison de toutes les gaussiennes photométriques représentant les pixels d'une image, la représentation de cette image change en fonction du type de modèle de gaussiennes utilisé.

La figure 4.4b montre des mélanges de gaussiennes de l'image du Yin et du Yang (Figure 4.4a) calculés avec le modèle de gaussienne 1 (eq. 4.1). De la même façon, la figure 4.4c montre des mélanges de gaussiennes de cette même image calculés avec le modèle de gaussienne 2 (eq. 4.2). Enfin, la figure 4.4d montre des mélanges de gaussiennes de l'image calculés avec le modèle de gaussienne trois (eq. 4.3). Pour les 3 modèles, les mélanges sont calculés en utilisant un paramètre d'extension λ_g de plus en plus grand.

Le fait de normaliser les gaussiennes (modèle 2) perturbe énormément la représentation de l'image sous forme de mélange de gaussiennes (Figure 4.4c). Contrairement aux modèles 1 et 3, même pour un λ_g relativement faible, le mélange obtenu avec le modèle 2 ne ressemble plus à l'image d'origine. Plus problématique encore, lorsque λ_g dépasse un certain seuil, les valeurs du mélange calculées avec ce modèle s'inversent par rapport à l'image. Les zones de l'image d'intensité basse deviennent des zones hautes dans le mélange et inversement. Ce phénomène est logique étant donné que les gaussiennes du modèle 2 sont normalisées. Puisque l'aire sous chaque gaussienne est égale à 1, lorsqu'une gaussienne est très étendue, son amplitude est alors très basse. Pour ces raisons, le modèle 2 n'est pas à privilégier pour nos travaux.

Les mélanges de gaussiennes obtenus avec les modèles 1 (Figure 4.4b) et 3 (Figure 4.4d) sont assez proches. Cependant, l'évolution des mélanges du modèle 3 en fonction du paramètre d'extension est plus progressive. De plus, lorsque le paramètre d'extension est élevé, le mélange calculé avec le modèle de gaussiennes

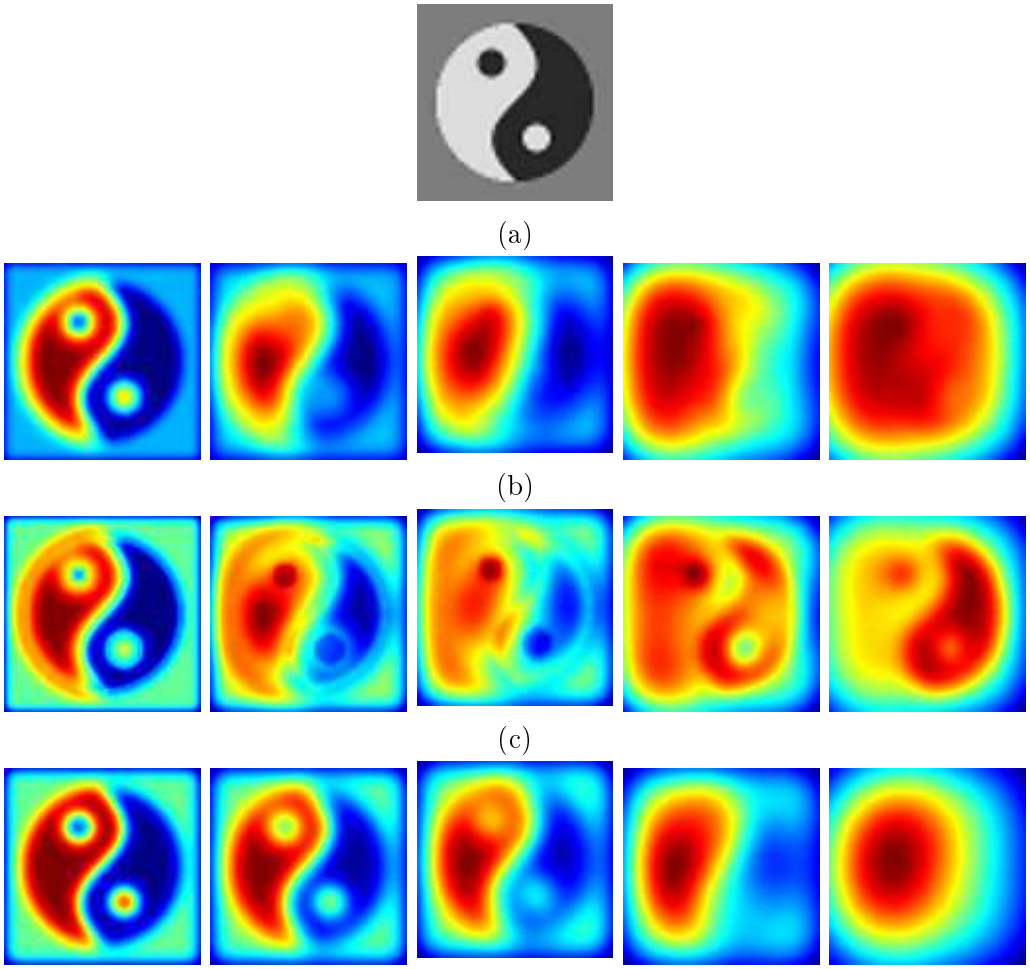


FIGURE 4.4: Influence du modèle sur les mélanges de gaussiennes : image d'origine (a), mélanges de gaussiennes de l'image d'origine calculés pour différents paramètres d'extension λ_g avec, respectivement, le modèle de gaussiennes 1 : Intensité/Envergure (b), 2 : Intensité/Envergure (Normalisé) (c), 3 : Intensité/Amplitude (d)

3 représente globalement l'image d'origine. Le modèle 1 reliant l'envergure des gaussiennes à l'intensité des pixels, lorsque le paramètre d'extension est élevé, les gaussiennes représentant des pixels d'intensité élevée sont très étendues et le mélange n'est plus représentatif de l'image d'origine.

Le modèle 3, reliant l'intensité d'un pixel à l'amplitude de sa gaussienne, semble le plus intéressant pour notre utilisation. En effet, contrairement aux deux autres modèles, pour un faible paramètre d'extension, les valeurs du mélange de gaussiennes sont identiques aux intensités des pixels en entrée. Plus généralement, quelque soit le paramètre d'extension, la forme globale du mélange de gaussiennes

reste proche du signal d'entrée pour le modèle 3 contrairement aux deux autres modèles. Qui plus est, les pixels purement noirs ($I(\mathbf{u}) = 0$) ne peuvent être pris en compte avec les modèles 1 et 2 car ils entraîneraient une division par 0 dans les équations (4.2) et (4.1). Pour ces différentes raison, dans la suite, nous privilégions l'utilisation du modèle de gaussienne 3 : Intensité/Amplitude.

4.3 Mélanges de gaussiennes comme caractéristiques visuelles denses

L'asservissement visuel virtuel basé mélanges de gaussiennes photométriques a pour objectif de minimiser la différence entre le mélange de gaussiennes d'une image numérique désirée $\mathbf{gm}(\widetilde{\mathbf{I}})$ et les mélanges de gaussiennes des images virtuelles générées dans le modèle 3D représentant l'environnement au cours de l'asservissement $\mathbf{gm}(\widetilde{\mathbf{I}_\mathbf{v}(\mathbf{r})})$. L'erreur $\mathbf{e}$ à minimiser s'écrit alors :

$$\mathbf{e} = \mathbf{gm}(\widetilde{\mathbf{I}}) - \mathbf{gm}(\widetilde{\mathbf{I}_\mathbf{v}}(\mathbf{r})) \tag{4.4}$$

où $\mathbf{r} = (t_x, t_y, t_z, \theta_{w_x}, \theta_{w_y}, \theta_{w_z})$ représente la pose courante de la caméra virtuelle dans le modèle. Comme pour l'asservissement visuel virtuel photométrique (Section 2.4), tous les pixels des images virtuelles ne contiennent pas forcément la projection d'un point 3D du modèle. C'est pourquoi, l'image virtuelle courante et l'image numérique désirée sont notées $\widetilde{\mathbf{I}_\mathbf{v}}(\mathbf{r})$ et $\widetilde{\mathbf{I}}$.

4.3.1 Loi de commande

La pose de la caméra à l'origine de l'image numérique que l'on souhaite atteindre est considérée comme la solution d'un problème d'optimisation non-linéaire, les 6 ddl de la caméra virtuelle ainsi que le paramètre d'extension des gaussiennes sont déterminés itérativement par une loi de contrôle de type Gauss-Newton :

$$\mathbf{v_g} = \mu \mathbf{L}^+_\mathbf{gm}(\widetilde{\mathbf{I}} - \widetilde{\mathbf{I}_\mathbf{v}}(\mathbf{r})) \tag{4.5}$$

où $\mathbf{v_g} = (\boldsymbol{\upsilon}\ \boldsymbol{\omega}\ \dot{\lambda}_g)^\top$ contient respectivement les vitesses linéaires et angulaires de la caméra et l'incrément du paramètre d'extension des gaussiennes. $\mathbf{L}^+_\mathbf{gm}$ est la pseudo-inverse de la matrice d'interaction liant les changements entre les mélanges de gaussiennes calculés sur les images $\mathbf{I_v}(\mathbf{r})$ acquises au cours de l'asservissement par rapport aux déplacements de la caméra. Enfin, le gain μ peut être utilisé pour régler la vitesse de convergence de la caméra.

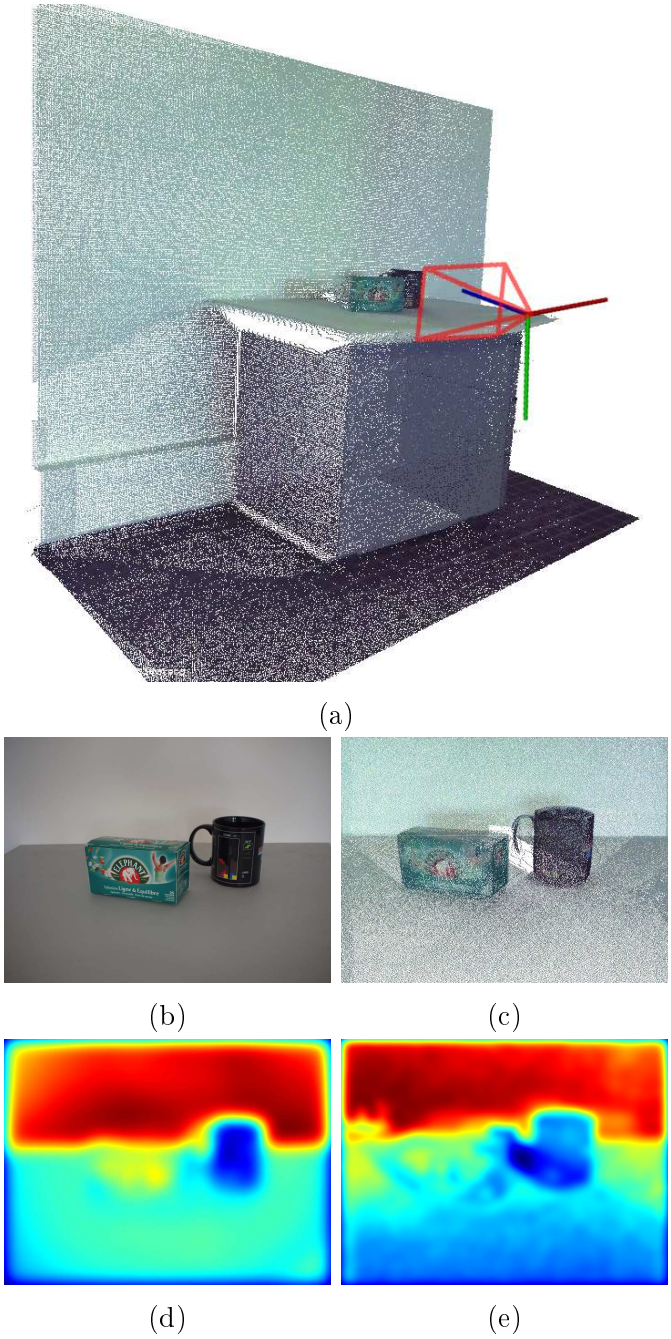
La matrice d'interaction $\mathbf{L_{gm}}$ est modélisée de la même manière que pour l'asservissement visuel basé mélanges de gaussiennes (Section 3.3) tout en prenant en considération les nuances inhérentes à l'utilisation d'images virtuelles générées dans un modèle 3D constitué de points (Section 2.4).

4.3.2 Calcul des gradients

La matrice d'interaction $\mathbf{L_{gm}}$ est composée, entre autres (Section 3.3.2), du gradient spatial de chaque point du mélange de gaussiennes courant. Lorsque l'information photométrique contenue dans les images est directement utilisée comme caractéristique visuelle dense, les gradients spatiaux $\vec{\nabla}\mathbf{I}$ (eq. 3.14) de l'image virtuelle courante $\mathbf{Iv}(\mathbf{r})$ sont les seules données nécessitant un traitement d'image (Section 2.4.2.3). Étant donné que nos images virtuelles résultent de la projection d'un modèle représentant l'environnement de type nuage de points (Figure 4.5a), certains pixels peuvent ne pas contenir la projection d'un des points 3D du nuage (Figure 4.5c). La prise en compte de ces pixels vides fausserait les gradients de l'image. Plus exactement, les gradients d'un pixel $\mathbf{u} = (u, v)$ de l'image sont calculés à partir de ces trois pixels voisins dans les 4 directions. C'est pourquoi, nous avons proposé de n'utiliser que les pixels de l'image qui contiennent la projection d'un point 3D du modèle et dont les douze pixels voisins contiennent également la projection d'un point 3D (Figure 2.8).

Dans le cas présent, nous modélisons les images par des mélanges de gaussiennes photométriques. Les gaussiennes étant définies de moins l'infini à plus l'infini, même si une image virtuelle possède des pixels ne contenant pas la projection d'un point 3D (Figure 4.5c), le mélange qui la représente est, quant à lui, continu et couvre la totalité de l'image (Figure 4.5e). Par conséquent, tous les points du mélange de gaussiennes peuvent être utilisés pendant l'asservissement visuel virtuel. De surcroît, la formulation des mélanges de gaussiennes photométriques modélisant les images est analytiquement connue. Ainsi, les gradients spatiaux des mélanges ne sont pas approximés par un traitement d'image mais peuvent être analytiquement exprimés.

Il est intéressant de noter que, dans l'exemple illustré en figure 4.5, l'image virtuelle générée à partir de la pose de la caméra virtuelle contient un grand nombre de pixels vides (Figure 4.5c) car la caméra est proche du nuage représentant la scène (Figure 4.5a). Dans cette configuration, si l'on exploite directement l'information photométrique des images comme caractéristiques visuelles, très peu de points sont alors utilisables. En revanche, malgré les pixels vides, le mélange de gaussiennes de l'image désirée (Figure 4.5d) et celui de l'image virtuelle (Figure 4.5e) sont très similaires.



(a) (b) (c) (d) (e)

FIGURE 4.5: Modèle composé de deux nuages de points 3D d'une scène simple (a), image numérique de la scène réelle (b), image virtuelle générée dans le modèle 3D (c), mélange de gaussiennes de l'image numérique (d), mélange de gaussiennes de l'image virtuelle (e)

4.4 Application

Nous utilisons cette nouvelle modélisation des images virtuelles et réelles comme caractéristiques visuelles denses d'asservissements visuels virtuels. Ces travaux s'inscrivent dans le cadre du projet E-Cathédrale, les expérimentations sont donc menées à l'intérieur et à l'extérieur de la cathédrale d'Amiens. Nous utilisons alors comme représentation virtuelle de l'environnement, le modèle 3D de la cathédrale composé par les nombreux nuages de points acquis au cours des campagnes d'acquisition du projet E-Cathédrale.

Comme avec l'asservissement visuel virtuel photométrique (Chapitre 2), nous souhaitons retrouver la pose à partir de laquelle une image numérique a été prise dans le but de projeter les couleurs de cette image sur le modèle 3D.

4.4.1 Colorisation photo-réaliste de nuages de points

Projeter les couleurs d'images numériques de bonne qualité prises à partir de poses judicieusement choisies sur des données 3D acquises par lasergrammétrie permet d'obtenir des représentations virtuelles géométriquement et visuellement fidèles d'un environnement réel. Pour cela, il faut être capable de recaler précisément les images numériques sur le modèle 3D. Ce recalage est une étape délicate et essentielle pour une colorisation de qualité.

4.4.1.1 Méthodologie

Nous avons proposé dans le chapitre 2 de formaliser ce problème de recalage comme une estimation de pose en deux étapes. La première étape consiste à estimer par asservissement visuel virtuel basé points (Section 2.3) une première pose de caméra virtuelle approximative. Cette pose est ensuite améliorée par asservissement visuel virtuel photométrique (Section 2.4). Cette démarche permet d'obtenir des résultats intéressants (Figure 2.12). Cependant, pour que le recalage final soit satisfaisant, il faut que la pose approximative estimée lors de la première étape soit suffisamment proche de la solution pour que la seconde étape permette à la caméra virtuelle d'atteindre la pose réelle. La première étape se basant sur des caractéristiques visuelles géométriques éparses, l'image numérique désirée et l'image virtuelle initiale doivent avoir un aspect visuel relativement proche pour être en mesure de détecter et de mettre en correspondance ces caractéristiques. Malgré l'homogénéisation des couleurs des nuages de points composant le modèle, la mise en correspondance est parfois difficile et ne permet pas d'estimer une pose approximative et donc de recaler l'image numérique sur le modèle 3D.

Les mélanges de gaussiennes photométriques permettent d'accroître significativement le domaine de convergence des asservissements visuels (Section 3.4).

Grâce à cette nouvelle modélisation des images, l'estimation d'une première pose proche n'est plus nécessaire. Dans le chapitre précédent, le paramètre d'extension λ_g^* des gaussiennes du mélange représentant l'image désirée était faible et constant tout au long de l'asservissement. Nous proposons ici d'initialiser l'asservissement avec un paramètre d'extension désiré et un paramètre d'extension initial élevés. Puis, pendant l'asservissement, lorsque l'erreur résiduelle (eq. 4.4) entre le mélange de gaussiennes désiré et le mélange de gaussiennes courant devient constante, le paramètre d'extension désiré est réduit. Le paramètre d'extension désiré est graduellement réduit de cette manière jusqu'à ce qu'il atteigne une valeur suffisamment faible pour que le mélange de gaussiennes désiré soit proche de l'image numérique désirée d'origine. Cette variante a pour but d'agrandir encore plus le domaine de convergence et de passer outre à l'aspect visuel dégradé des images virtuelles. En effet, le fait de commencer l'asservissement avec deux grands paramètres d'extension permet d'obtenir des mélanges de gaussiennes désirés et initiaux qui représentent respectivement la scène vue dans l'image numérique réelle et dans l'image virtuelle de façon globale. Ainsi, l'asservissement visuel virtuel recale tout d'abord globalement la scène puis de plus en plus rigoureusement jusqu'à atteindre la précision à convergence que l'on obtiendrait avec une caractéristique visuelle purement photométrique.

Si la caméra à l'origine des images numériques est calibrée, alors la caméra virtuelle est modélisée avec les paramètres intrinsèques (eq. 1.3) de la caméra réelle. Si ce n'est pas le cas, les paramètres intrinsèques de la caméra virtuelle sont, dans un premier temps, manuellement choisis. Puis, lorsque l'estimation de pose converge, les paramètres intrinsèques sont optimisés au même titre que les paramètres extrinsèques. Durant cette phase, le paramètre d'extension désiré est constant et de valeur faible.

La méthode de colorisation employée est identique à celle utilisée précédemment. Les points visibles dans une première image numérique sont colorisés et les autres sont marqués comme non-colorés. Les points visibles dans la deuxième image numérique et marqués comme étant non-colorés sont alors colorisés. L'opération est répétée sur l'ensemble des images numériques ou jusqu'à ce que la totalité des points du nuage soit colorisée.

4.4.1.2 Résultat

La chaire de vérité -

Cette nouvelle approche est utilisée pour colorer le modèle 3D de la chaire de vérité de la cathédrale d'Amiens. La figure 4.7 montre trois images numériques réelles de cette oeuvre.

La première image numérique utilisée pour coloriser le modèle 3D est la prise de vue par la gauche de la chaire (Figure 4.6a). La caméra à l'origine des images

(a) (b) (c)

FIGURE 4.6: Images numériques la chaire de vérité : la prise de vue de gauche (a) et de droite (c) de la chaire sont utilisées pour la colorisation du modèle. La prise de vue de face (b) est utilisée comme évaluation qualitative de la colorisation

numériques n'est pas calibrée. Les paramètres intrinsèques de la caméra virtuelle sont, dans un premier, choisis manuellement, puis ils seront estimés à convergence.

Pour initialiser l'asservissement visuel virtuel basé mélanges de gaussiennes photométriques, la caméra virtuelle est grossièrement placée dans le modèle afin que la chaire de vérité soit dans le champ de vue de la caméra. La figure 4.7a montre l'image $\mathbf{I_v}(\mathbf{r_0})$ où $\mathbf{r}_0$ représente la pose initiale de la caméra virtuelle, autrement dit, à l'itération 0 de l'optimisation. La figure 4.7d montre le mélange de gaussiennes calculé sur cette image initiale avec un paramètre d'extension λ_g de 18.0 choisi expérimentalement. Le mélange de gaussiennes de la figure 4.7g est calculé à partir de l'image numérique désirée (Figure 4.6a) avec un paramètre d'extension λ_g^* égal à λ_g. Ensuite, au cours de l'asservissement, le paramètre d'extension désiré λ_g^* est réduit graduellement à chaque fois que l'erreur résiduelle devient constante. Les figures 4.7b, 4.7e, 4.7h montrent respectivement l'image virtuelle générée par la caméra au milieu de l'asservissement, le mélange de gaussiennes qui lui est associé et le mélange de gaussiennes de l'image désirée. À ce stade, on peut voir que le paramètre d'extension désiré λ_g^* a été réduit à 5.0. Enfin, les figures 4.7c, 4.7f, 4.7i montrent respectivement l'image virtuelle générée à convergence, le mélange de gaussiennes qui lui est associé et le mélange de gaussiennes de l'image désirée. À convergence, le paramètre d'extension désiré λ_g^* est très faible $\lambda_g^* = 0.5$, les mélanges de gaussiennes sont alors très proches de l'image virtuelle et de l'image réelle d'origine.

L'image virtuelle générée à la fin de l'estimation de pose (Figure 4.7c) montre

que, même en partant d'une pose de caméra très éloignée, la caméra virtuelle a convergé vers la pose de caméra à l'origine de l'image numérique désirée (Figure 4.6a).

Nous pouvons observer que l'erreur entre le mélange de gaussiennes désiré et le mélange de gaussiennes calculé sur les images virtuelles acquises tout au long de l'asservissement est correctement minimisée et décroit au fur et à mesure des réductions du paramètre d'extension désiré (Figure 4.8). Malgré les déplacements importants entre la pose initiale et la pose désirée, la caméra s'est virtuellement déplacée de plusieurs mètres pour atteindre la pose finale (Figure 4.8b).

À la différence des expérimentations menées dans le chapitre 3, les images courantes et l'image désirée sont de nature différente. Il n'y a pas de raison particulière à ce que le paramètre d'extension courant λ_g utilisé pour calculer les mélanges de gaussiennes des images virtuelles converge parfaitement vers le paramètre d'extension désirée λ_g^*. Cependant, l'évolution du paramètre d'extension λ_g par rapport aux réductions du paramètre d'extension désirée λ_g^* est cohérente (Figure 4.8a).

À ce stade, la pose de la caméra virtuelle est très proche de la pose de la caméra à l'origine de l'image numérique désirée. Cependant, les paramètres intrinsèques de la caméra virtuelle n'étant pas les mêmes que ceux de la caméra réelle, des décalages sont toujours présents entre l'image désirée et l'image virtuelle générée à convergence. Ces décalages sont mis en évidence sur la figure 4.9a. Sur cette figure, les contours de l'image numérique désirée sont superposés à l'image virtuelle obtenue à convergence. Les décalages entre les deux images sont ainsi plus facilement visualisables.

Pour corriger ces décalages, nous effectuons une nouvelle fois un asservissement visuel virtuel basé mélanges de gaussiennes photométriques, en partant cette fois de la pose de caméra estimée précédemment. Ici, en plus des six degrés de liberté de la caméra et du paramètre d'extension des gaussiennes, nous estimons également les quatre paramètres intrinsèques de la caméra. Durant cet asservissement, le paramètre d'extension désiré est constant et très faible. L'estimation des paramètres intrinsèques de la caméra à l'origine de l'image numérique désirée permet d'affiner le recalage. Comme le montre la figure 4.9b, après cette étape les contours de l'image désirée suivent parfaitement la composition de la scène projetée dans l'image virtuelle après convergence des paramètres extrinsèques et intrinsèques de la caméra virtuelle.

Le même procédé est employé pour estimer la pose de caméra d'une seconde image numérique (Figure 4.6c). Les couleurs des deux images numériques dont les poses de caméra ont été estimées sont projetées sur le modèle 3D de la chaire de vérité. La figure 4.10a montre le nuage de points de la chaire dont les couleurs sont celles acquises par le scanner laser. La figure 4.10b, quant à elle, montre le nuage de points de la chaire après la colorisation.

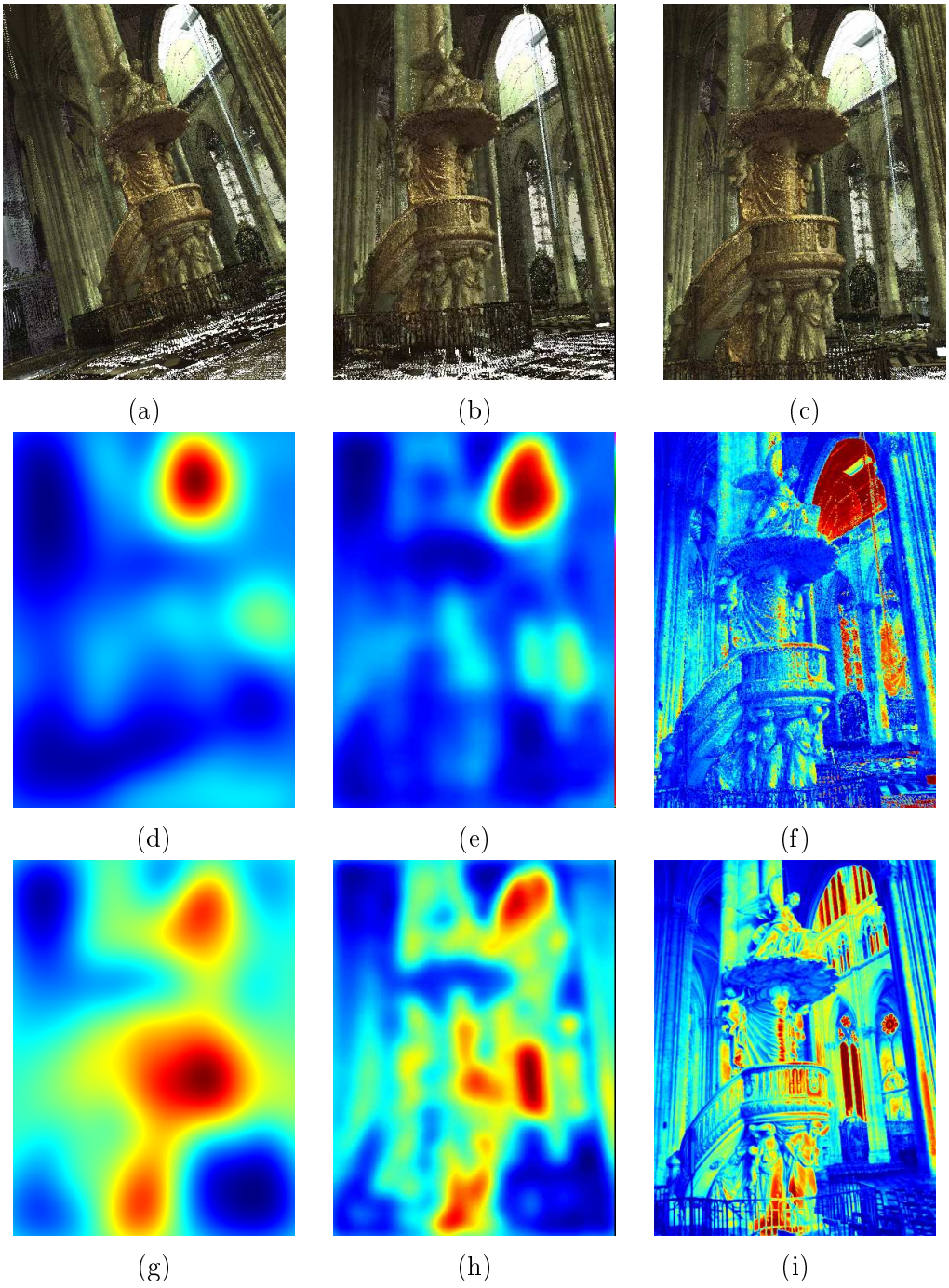
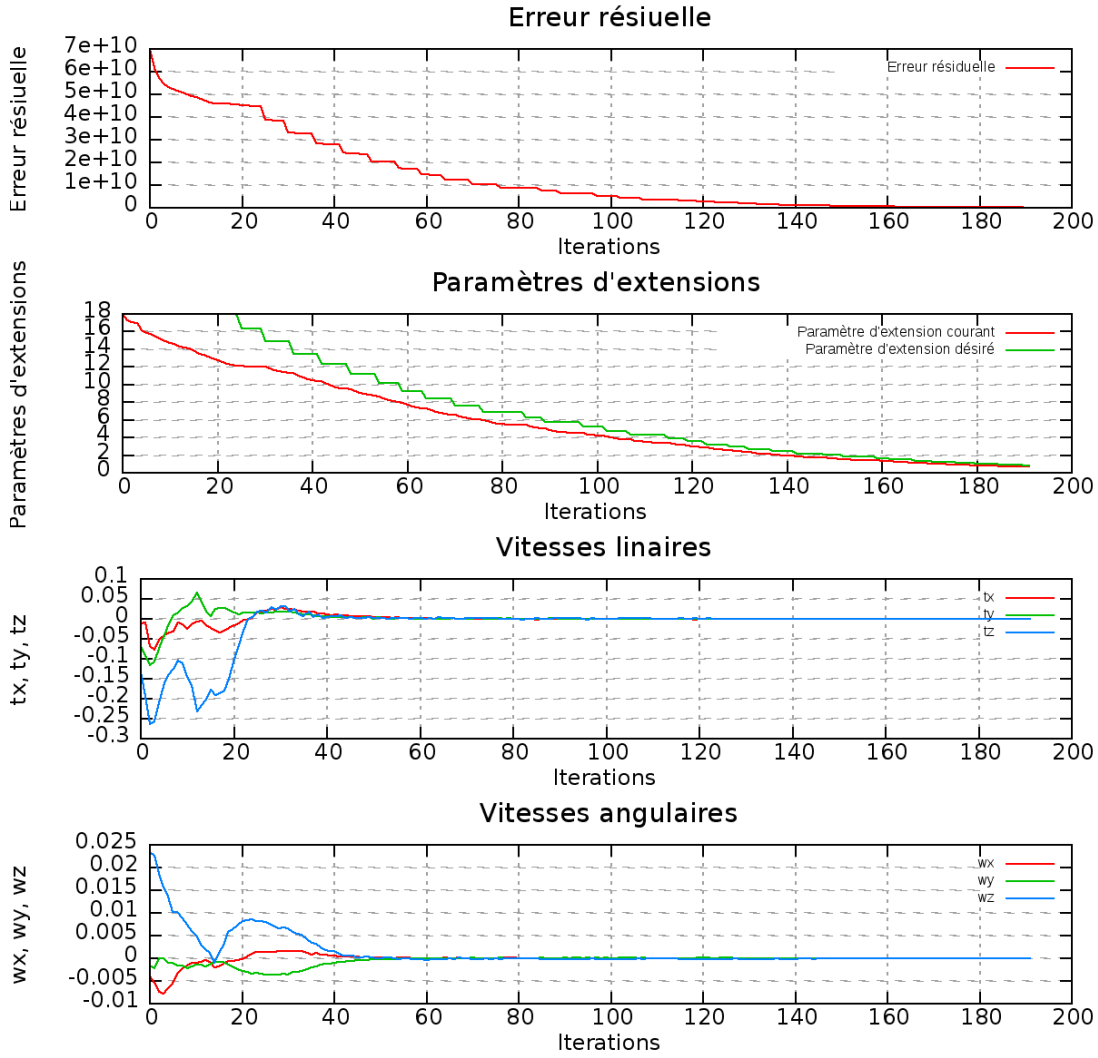
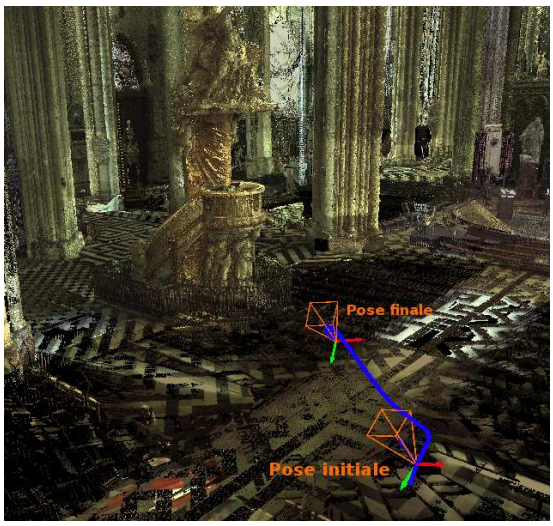


FIGURE 4.7: Asservissement visuel virtuel basé mélanges de gaussiennes photométriques : images virtuelles générées respectivement à la pose de caméra initiale, au milieu de l'asservissement et à convergence (a-c), mélanges de gaussiennes calculés à partir des images virtuelles (b-f) et mélanges de gaussiennes calculés à partir de l'image numérique désirée (Figure 4.6a)

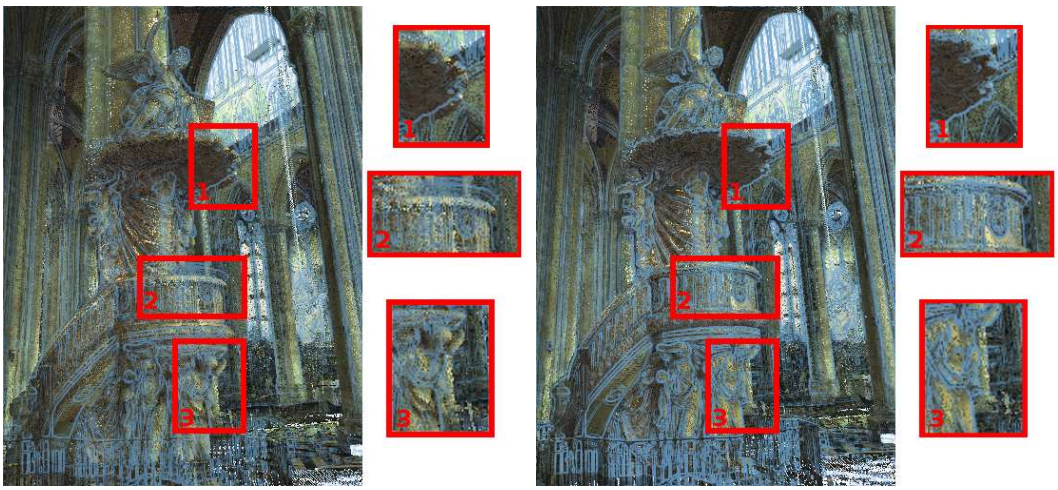


(a)



(b)

FIGURE 4.8: Évolution de l'erreur entre le mélange de gaussiennes désiré et les mélanges de gaussiennes calculés sur les images virtuelles, des paramètres d'extension désiré et courant et des vitesses envoyées à la caméra (a). Trajectoire empruntée par la caméra virtuelle durant l'asservissement (b)



(a) (b)

FIGURE 4.9: Correction des décalages inhérents à la non-connaissance des paramètres intrinsèques de la caméra à l'origine de l'image numérique désirée : contours de l'image numérique en bleu superposés à l'image virtuelle à convergence avant l'estimation des paramètres intrinsèques (a), et après (b)

(a)

(b)

FIGURE 4.10: Résultat de la colorisation : nuages de points de la chaire de vérité avant colorisation (a), et après (b)

4.5 Conclusion

Dans ce chapitre, nous avons montré qu'il est possible de résoudre le problème d'estimation de pose de caméra sous le formalisme de l'asservissement visuel virtuel en utilisant les mélanges de gaussiennes comme caractéristiques visuelles denses. Ce chapitre fait la jonction entre les contributions du deuxième et du troisième chapitres.

Les gaussiennes 2D des mélanges peuvent prendre différents jeux de paramètres pour représenter les pixels des images. Plusieurs modèles de gaussiennes ont été proposés et leurs influences sur les mélanges ont été comparées. Un modèle différent de celui utilisé dans le chapitre 3 s'est démarqué, correspondant davantage au concept initial du pouvoir d'attraction attribué aux pixels des images.

Les mélanges de gaussiennes se sont révélés être des caractéristiques visuelles très intéressantes pour les modèles 3D utilisés dans nos travaux, c'est-à-dire issus de prélèvements par scanners laser. En effet, étant donné que les mélanges de gaussiennes photométriques permettent d'élargir significativement le domaine de convergence des asservissements, l'utilisation d'un premier calcul de pose basé points n'est plus nécessaire pour initialiser le calcul de pose dense comme c'était le cas avec l'asservissement visuel virtuel purement photométrique. Les mélanges de gaussiennes ont également l'avantage d'atténuer la mauvaise qualité, d'un point de vue photométrique, des images virtuelles.

Les mélanges de gaussiennes photométriques se prêtent tout particulièrement bien aux images virtuelles générées à partir de nuages de points colorés. Les gaussiennes étant définies de moins l'infini à plus l'infini, même si une image virtuelle possède des pixels ne contenant pas la projection d'un point 3D, le mélange qui la représente est, quant à lui, continu et couvre la totalité de l'image. Par conséquent, tous les échantillons du mélange de gaussiennes peuvent être utilisés pendant l'asservissement visuel virtuel.

Nous avons également proposé dans ce chapitre de modifier graduellement l'extension des gaussiennes du mélange calculé sur l'image désirée. Cela permet d'agrandir encore plus le domaine de convergence de l'asservissement visuel. Les résultats de colorisation du modèle 3D montrent que la précision à convergence de l'asservissement reste, quant à elle, excellente.

Chapitre 5

Conclusion et perspectives

Ce mémoire de thèse traite les questions d'estimation de pose et de positionnement de robot basés vision. Il prend le parti de les aborder sous le formalisme des asservissements visuels, dans le cadre duquel de nouvelles contributions sont proposées.

Tout d'abord, nous avons énoncé quelques notations et notions fondamentales sur la vision par ordinateur. Nous avons introduit les différentes modélisations de caméra utilisées dans cette thèse ainsi que les relations géométriques entre la caméra et l'environnement dans lequel elle évolue. Nous avons également dressé un état-de-l'art non-exhaustif répertoriant les différentes caractéristiques visuelles employées en asservissement visuel. Enfin, nous avons présenté les différents types de modèles 3D exploités en asservissement visuel virtuel.

Dans le deuxième chapitre, nous avons montré qu'il est possible de résoudre le problème d'estimation de pose de caméra, sous le formalisme de l'asservissement visuel virtuel, en utilisant directement les caractéristiques photométriques d'images réelles et virtuelles. Nous avons tout d'abord présenté le projet E-Cathédrale, le programme de recherche dans lequel s'inscrivent les travaux de cette thèse. Nous nous sommes attardés sur la lasergrammétrie, la méthode de numérisation qui a permis d'acquérir les nuages de points colorés utilisés durant nos expérimentations.

Les modèles provenant d'acquisitions par scanners laser peuvent contenir un très grand nombre de mesures 3D et sont, généralement, composés de plusieurs nuages de points colorés avec des teintes différentes, ce qui génère des incohérences visuelles. Pour permettre l'exploitation de ce type de modèle dans de meilleures conditions, nous avons proposé des méthodes de pré-traitement des nuages de points colorés composant le modèle complet. La première est une structuration spatiale du modèle qui permet de n'utiliser que les points utiles du modèle en fonction de la position de la caméra pour générer les images virtuelles. En développant plus en profondeur cette approche, nous pensons que cela pourrait donner naissance à une méthode de visualisation temps-réel de ce type d'environnement virtuel. Le second pré-traitement permet d'homogénéiser les couleurs des nuages de points afin de corriger l'aspect visuel du modèle et, par extension, des images virtuelles générées. L'homogénéisation des couleurs rend possible la détection et la mise en correspondance de primitives géométriques

entre les images virtuelles et les images réelles de la scène. Des calculs de pose par asservissement visuel virtuel basé points sont alors réalisables et nous avons proposé de les utiliser pour initialiser le calcul de pose photométrique. Une étude visant à déterminer quel type de modèle et quel critère de similarité sont les plus pertinents a été réalisée. Puis, une formulation de l'asservissement visuel virtuel photométrique avec comme représentation virtuelle de l'environnement un modèle composé de nuages de points colorés a été proposée. Les différentes expérimentations, que ce soit pour la colorisation du modèle ou pour la localisation de robot mobile, ont permis de valider l'approche et les limitations de la méthode, principalement liées au domaine de convergence limité des asservissements visuels photométriques, ont été mis en évidence.

Les limites des asservissements visuels photométriques mises en lumière dans les expérimentations de calcul de pose du deuxième chapitre ont encouragé nos recherches vers une nouvelle modélisation des images permettant d'accroître le domaine de convergence des asservissements. Dans un premier temps, cette nouvelle modélisation des images a été introduite dans le troisième chapitre comme caractéristique visuelle d'asservissement visuel. L'idée générale derrière cette modélisation d'image est de vouloir attribuer à chaque pixel de l'image un pouvoir d'attraction en représentant chaque pixel par une fonction gaussienne 2D. L'image complète est alors obtenue en combinant toutes les gaussiennes sous la forme d'un mélange de gaussiennes. En faisant varier l'envergure des gaussiennes, il est possible de passer d'une représentation globale de l'image à un mélange dont les valeurs sont proches des intensités de l'image modélisée. Nous avons proposé alors d'optimiser, en plus des degrés de liberté de la caméra, l'extension des gaussiennes du mélange modélisant l'image courante pendant l'asservissement. La matrice d'interaction a été développée pour cette nouvelle modélisation des images. Une simple loi de commande du premier ordre est suffisante pour réaliser le contrôle des degrés de liberté de la caméra et pour estimer l'évolution de l'extension des gaussiennes. De nombreuses simulations et expérimentations réelles ont été menées afin de valider cette nouvelle approche. Le choix de l'extension des gaussiennes du mélange de l'image désirée et celle du mélange de l'image initiale a été discuté et nous avons démontré qu'il est possible de commencer l'asservissement avec des pixels bénéficiant d'un fort pouvoir d'attraction et de le finir avec des mélanges représentant les images sous une forme quasiment photométrique. Les mélanges de gaussiennes photométriques permettent ainsi d'élargir significativement le domaine de convergence de l'asservissement tout en conservant la grande précision à convergence des asservissements photométriques.

Le quatrième chapitre fait la jonction entre les contributions du deuxième et du troisième chapitre. Nous avons montré qu'il est possible de résoudre le problème d'estimation de pose de caméra sous le formalisme de l'asservissement visuel virtuel en utilisant les mélanges de gaussiennes comme caractéristiques vi-

suelles denses. Nous avons étudié et comparé différents jeux de paramètres pour représenter les pixels des images sous forme de gaussiennes 2D. Un modèle de gaussiennes différent de celui utilisé dans le troisième chapitre s'est révélé plus cohérent avec le concept de base des mélanges de gaussiennes photométriques. Les mélanges de gaussiennes se sont révélés être des caractéristiques visuelles très intéressantes pour les modèles 3D utilisés dans nos travaux, c'est-à-dire les modèles composés de nuages de points colorés. En effet, étant donné que les mélanges de gaussiennes photométriques permettent d'élargir significativement le domaine de convergence des asservissements, l'utilisation d'un premier calcul de pose basé points n'est plus nécessaire pour initialiser le calcul de pose dense comme c'était le cas dans le deuxième chapitre avec l'asservissement visuel virtuel purement photométrique. Les mélanges de gaussiennes photométriques se prêtent tout particulièrement bien aux images virtuelles générées à partir de nuages de points colorés. Les gaussiennes étant définies de moins l'infini à plus l'infini, même si une image virtuelle possède des pixels ne contenant pas la projection d'un point 3D, le mélange qui la représente est, quant à lui, continu et couvre la totalité de l'image. Par conséquent, tous les échantillons du mélange de gaussiennes peuvent être utilisés pendant l'asservissement visuel virtuel. Qui plus est, la mauvaise qualité, d'un point de vue photométrique, des images virtuelles est atténuée lorsqu'elles sont représentées par des mélanges de gaussiennes. Nous avons également proposé de faire évoluer graduellement l'extension des gaussiennes du mélange représentant l'image désirée. Cela permet d'agrandir encore plus le domaine de convergence de l'asservissement visuel. Comme l'ont montré les résultats de colorisation du modèle 3D, la précision à convergence de l'asservissement reste, quant à elle, excellente.

À l'issue des travaux présentés dans ce manuscrit, plusieurs perspectives de recherche sont envisageables. Tout d'abord, une étude de stabilité de l'asservissement visuel basé mélanges de gaussiennes photométriques devrait être menée. Même si les mélanges de gaussiennes photométriques permettent de faire converger la caméra en partant d'une pose très éloignée de la solution et que les résultats ont montré une précision à convergence très précise, la méthode n'a été évaluée qu'expérimentalement.

Actuellement, nous n'avons utilisé qu'un seul modèle de gaussiennes en asservissement visuel. Il serait intéressant d'exploiter le modèle de gaussiennes qui s'est démarqué pour l'asservissement visuel virtuel dans le quatrième chapitre. Pour aller plus loin, il faudrait comparer plus en détails l'influence des différents modèles de gaussiennes et étudier le lien entre la grandeur de l'extension des gaussiennes et la trajectoire de la caméra durant les asservissements visuels en général. Une méthode permettant d'initialiser automatiquement le paramètre d'extension désiré et le paramètre d'extension initial des mélanges serait égale-

ment une perspective de recherche à aborder.

Le choix de représenter les pixels des images par des gaussiennes 2D est discutable et d'autres fonctions 2D peuvent être utilisées. Une seconde perspective de recherche pourrait être d'étudier d'autres caractéristiques visuelles denses en représentant les pixels par des fonctions 2D ayant une distribution différente et, pourquoi pas, davantage de paramètres à optimiser pendant l'asservissement.

Le principal défaut des mélanges de gaussiennes reste leur temps de calcul élevé. Qui plus est, l'algorithme de calcul des mélanges est difficilement parallélisable. Il devient alors intéressant d'entreprendre des recherches dans le but de créer une méthode d'approximation rapide du mélange de gaussiennes représentant une image.

Nous avons exploité les mélanges de gaussiennes comme caractéristiques visuelles denses pour l'estimation de pose de caméra pour recaler des images numériques quelconques sur des nuages de points 3D, afin d'en améliorer la qualité des couleurs. Au vu des résultats obtenus, l'utilisation des mélanges de gaussiennes photométriques gagnerait à être étendue à d'autres types d'applications, comme, par exemple, la localisation de robot mobile. Nous avons vu dans le deuxième chapitre que l'asservissement visuel virtuel photométrique est mis en échec dès l'instant où deux images successivement acquises par le robot sont trop différentes. Les mélanges de gaussiennes photométriques élargissant significativement le domaine de convergence des asservissements visuels, leurs utilisations devraient apporter une solution adaptée à ces problèmes.

Pour l'avenir, j'aimerais étendre l'utilisation des mélanges de gaussiennes photométrique en variant les applications, soit en changeant de thème d'application, en abordant, par exemple, la réalité augmentée, soit en restant en robotique mais en allant plus loin, comme par exemple la robotique multi-tâche ou encore en travaillant sur des robots aériens. Mais globalement, je souhaite rester impliqué dans la vision artificielle et le traitement d'images pour la robotique car ces thèmes de recherche présentent encore de grands défis à mes yeux

Bibliographie

[Abmayr 2004] T. Abmayr, F. Härtl, M. Mettenleiter, A. Heinz, B. Neumann et C. Fröhlich. *Realistic 3D Reconstruction - Combining Laserscan Data with RGB Color Information.* XXth ISPRS Congress : Proceedings of Commission V, 2004. (Cité en page 60.)

[Adan 2012] A. Adan, P. Merchan et S. Salamanca. *Creating Realistic 3D Models From Scanners by Decoupling Geometry and Texture.* International Conference on Pattern Recognition (ICPR), pages 457–460, 2012. (Cité en page 60.)

[Alshawabkeh 2004] Y. Alshawabkeh et N. Haala. *Integration of Digital Photogrammetry and Laser Scanning for Heritage Documentation.* IAPRS, pages 12–23, 2004. (Cité en page 60.)

[Andreff 2002] Nicolas Andreff, Bernard Espiau et Radu P. Horaud. *Visual Servoing from Lines.* Int. J. Robot. Res., vol. 21, no. 8, pages 679–700, 2002. (Cité en page 20.)

[Ardouin 2013] J. Ardouin, A. Lecuyer, M. Marchal et E. Marchand. *Navigating in Virtual Environments with 360o Omnidirectional Rendering.* In IEEE Symp. on 3D User Interfaces, 3DUI 2013, pages 95–98, Orlando, USA, March 2013. (Cité en page 16.)

[Bakthavatchalam 2013] M. Bakthavatchalam, F. Chaumette et E. Marchand. *Photometric moments : New promising candidates for visual servoing.* In IEEE Int. Conf. on Robotics and Automation, ICRA'13, pages 5521–5526, Karlsruhe, Germany, May 2013. (Cité en pages 31 et 32.)

[Bakthavatchalam 2015] M. Bakthavatchalam, F. Chaumette et O. Tahri. *An Improved Modelling Scheme for Photometric Moments with Inclusion of Spatial Weights for Visual Servoing with Partial Appearance/Disappearance.* In IEEE Int. Conf. on Robotics and Automation, ICRA'15, pages 6037–6043, Seattle, WA, May 2015. (Cité en page 32.)

[Barreto 2001] João P. Barreto et Helder Araújo. *Issues on the Geometry of Central Catadioptric Image Formation.* In Computer Society Conference on Computer Vision and Pattern Recognition, pages 422–427, 2001. (Cité en pages 11, 13 et 21.)

[Barreto 2004] João Pedro de Almeida Barreto. *General central projection systems : Modeling, calibration and visual servoing.* 2004. (Cité en page 21.)

[Bateux 2015] Quentin Bateux et Eric Marchand. *Direct visual servoing based on multiple intensity histograms.* In IEEE Int. Conf. on Robotics and Au-

tomation, ICRA'15, Seattle, United States, Mai 2015. (Cité en pages 28 et 29.)

[Belkhouche 2012] Yassine Belkhouche, Bill Buckles, Prakash Duraisamy et Kamesh Namuduri. *Registration of 3D-LiDAR Data With Visual Imagery Using Shape Matching.* Int. Conf. on Image Processing, Computer Vision, and Pattern Recognition, pages 749–754, 2012. (Cité en page 60.)

[Benhimane 2004] Selim Benhimane et Ezio Malis. *Real-time image-based tracking of planes using efficient second-order minimization.* IEEE IROS, pages 943–948, 2004. (Cité en page 24.)

[Benhimane 2007] S. Benhimane et E. Malis. *Homography-based 2D Visual Tracking and Servoing.* Int. J. Rob. Res., vol. 26, no. 7, pages 661–676, jul 2007. (Cité en page 24.)

[Besl 1992] Paul J. Besl et Neil D. McKay. *A Method for Registration of 3-D Shapes.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 14, no. 2, pages 239–256, 1992. (Cité en page 44.)

[Caron 2010a] G. Caron, E. Marchand et E. Mouaddib. *Omnidirectional Photometric Visual Servoing.* In IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, IROS'10, pages 6202–6207, Taipei, Taiwan, Taiwan, 2010. (Cité en page 24.)

[Caron 2010b] G. Caron, E. Marchand et E.M. Mouaddib. *Omnidirectional photometric visual servoing.* pages 6202–6207, Oct 2010. (Cité en page 25.)

[Caron 2012] G. Caron, E. Mouaddib et E. Marchand. *3D model based tracking for omnidirectional vision : a new spherical approach.* Robotics and Autonomous Systems, vol. 60, no. 8, pages 1056–1068, August 2012. (Cité en page 35.)

[Caron 2013] Guillaume Caron, Dominique Leclet-Groux et El Mustapha Mouaddib. *From Heritage Building Digitization To Computerized Education.* pages 1–1, 2013. (Cité en page 63.)

[Caron 2014] G. Caron, A. Dame et E. Marchand. *Direct model-based visual tracking and pose estimation using mutual information.* Image and Vision Computing, vol. 32, no. 1, pages 54–63, January 2014. (Cité en pages 35 et 36.)

[Cha 2002] Sung-Hyuk Cha et Sargur N. Srihari. *On measuring the distance between histograms.* Pattern Recognition, vol. 35, no. 6, pages 1355–1370, 2002. (Cité en page 29.)

[Chaumette 1990] F. Chaumette. La relation vision-commande : Theorie et application a des taches robotiques. 1990. (Cité en page 21.)

[Chaumette 2004] F. Chaumette. *Image moments : a general and useful set of features for visual servoing*. IEEE Trans. on Robotics, vol. 20, no. 4, pages 713–723, August 2004. (Cité en page 31.)

[Chaumette 2006] F. Chaumette et S. Hutchinson. *Visual servo control, Part I : Basic approaches*. IEEE Robotics and Automation Magazine, vol. 13, no. 4, pages 82–90, December 2006. (Cité en pages 19 et 20.)

[Chaumette 2007] François Chaumette et S. Hutchinson. *Visual servo control, Part II : Advanced approaches*. IEEE Robotics and Automation Magazine, vol. 14, no. 1, pages 109–118, 2007. (Cité en page 22.)

[Collewet 2008] C. Collewet, E. Marchand et F. Chaumette. *Visual servoing set free from image processing*. In IEEE Int. Conf. on Robotics and Automation, ICRA'08, pages 81–86, Pasadena, California, May 2008. (Cité en pages 24, 26, 51, 57 et 100.)

[Comport 2003a] A.I. Comport, E. Marchand et F. Chaumette. *A real-time tracker for markerless augmented reality*. pages 36–45, October 2003. (Cité en pages 34 et 71.)

[Comport 2003b] A.I. Comport, M. Pressigout, E. Marchand et F. Chaumette. *A Visual Servoing Control Law that is Robust to Image Outliers*. vol. 1, pages 492–497, October 2003. (Cité en page 26.)

[Comport 2006] A.I. Comport, E. Marchand, M. Pressigout et F. Chaumette. *Real-time markerless tracking for augmented reality : the virtual visual servoing framework*. IEEE Trans. on Visualization and Computer Graphics, vol. 12, no. 4, pages 615–628, July 2006. (Cité en page 34.)

[Corsini 2009] Massimiliano Corsini, Matteo Dellepiane, Federico Ponchio et Roberto Scopigno. *Image-to-Geometry Registration : a Mutual Information Method exploiting Illumination-related Geometric Properties*. Computer Graphics Forum, vol. 28, no. 7, pages 1755–1764, 2009. (Cité en pages 36, 60 et 61.)

[Corsini 2012] M. Corsini, M. Dellepiane, F. Ganovelli, R. Gherardi, a. Fusiello et R. Scopigno. *Fully Automatic Registration of Image Sets on Approximate Geometry*. International Journal of Computer Vision, vol. 102, no. 1-3, pages 91–111, Août 2012. (Cité en page 60.)

[Courbon 2007] J. Courbon, Y. Mezouar, L. Eckt et P. Martinet. *A generic fisheye camera model for robotic applications*. pages 1683–1688, Oct 2007. (Cité en page 11.)

[Dahmouche 2012] Redwan Dahmouche, Nicolas Andreff, Youcef Mezouar, Omar Ait-Aider et Philippe Martinet. *Dynamic visual servoing from sequential regions of interest acquisition*. I. J. Robotic Res., vol. 31, no. 4, pages 520–537, 2012. (Cité en page 20.)

[Dalal 2005] Navneet Dalal et Bill Triggs. *Histograms of Oriented Gradients for Human Detection*. In Cordelia Schmid, Stefano Soatto et Carlo Tomasi, editeurs, International Conference on Computer Vision & Pattern Recognition, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005. (Cité en page 28.)

[Dame 2010] A. Dame et E. Marchand. *Improving mutual information based visual servoing*. pages 5531–5536, May 2010. (Cité en page 27.)

[Dame 2012] A. Dame et E. Marchand. *Second order optimization of mutual information for real-time image registration*. IEEE Trans. on Image Processing, vol. 21, no. 9, pages 4190–4203, September 2012. (Cité en page 27.)

[Deguchi 2000] Koichiro Deguchi. *A Direct Interpretation of Dynamic Images with Camera and Object Motions for Vision Guided Robot Control*. International Journal of Computer Vision, vol. 37, no. 1, pages 7–20, 2000. (Cité en page 24.)

[Delabarre 2012] B. Delabarre et E. Marchand. *Visual Servoing using the Sum of Conditional Variance*. pages 1689–1694, October 2012. (Cité en page 27.)

[Dionnet 2007] Fabien Dionnet et Eric Marchand. *Robust stereo tracking for space applications*. pages 3373–3378, 2007. (Cité en page 34.)

[Espiau 1992] B. Espiau, F. Chaumette et P. Rives. *A new approach to visual servoing in robotics*. IEEE J RA, vol. 8, no. 3, pages 313–326, 1992. (Cité en page 20.)

[Feddema 1989] J. T. Feddema, C. S. G. Lee et O. R. Mitchell. *Automatic Selection of Image Features for Visual Servoing of a Robot Manipulator*. In Proc. of the 1989 IEEE International Conference on Robotics and Automation (Vol. 2), pages 832–837, Scottsdale, AZ, 1989. (Cité en page 21.)

[Geyer 2000] Christopher Geyer et Kostas Daniilidis. *A unifying theory for central panoramic systems and practical implications*. In In ECCV, pages 445–461, 2000. (Cité en page 11.)

[Gratal 2011] X. Gratal, J. Romero et D. Kragic. *Virtual Visual Servoing for Real-Time Robot Pose Estimation*. In International Federation of Automatic Control World Congress, IFAC, 2011. (Cité en page 35.)

[Gratal 2013] Xavi Gratal, Christian Smith, Moarten Bjorkman et Danica Kragic. *Integrating 3D Features and Virtual Visual Servoing for Hand-Eye and Humanoid Robot Pose Estimation*. In IEEE-RAS International Conference on Humanoid Robots, pages 240–245, 2013. (Cité en page 35.)

[Habibi 2014] Z. Habibi, G. Caron et E. Mouaddib. *3D model automatic : exploration Smooth and Intelligent Virtual Camera Control*. November 2014. (Cité en page 63.)

[Hadj-Abdelkader 2010] Hicham Hadj-Abdelkader, Youcef Mezouar et Philippe Martinet. *Points based visual servoing with central cameras.* vol. 401/2010, pages 295–313, 2010. (Cité en page 21.)

[Horn 1980] Berthold K.P. Horn et Brian G. Schunck. *Determining Optical Flow.* Rapport technique, Cambridge, MA, USA, 1980. (Cité en pages 25, 57 et 85.)

[Huber 1981] Peter J. Huber. *Robust Statistics.* 1981. (Cité en page 27.)

[Hutchinson 1996] Seth Hutchinson, Greg Hager et Peter Corke. *A Tutorial on Visual Servo Control.* IEEE Transactions on Robotics and Automation, vol. 12, pages 651–670, 1996. (Cité en page 19.)

[Kallem 2007] Vinutha Kallem, Maneesh Dewan, John P. Swensen, Gregory D. Hager et Noah J. Cowan. *Kernel-based visual servoing.* In IROS, pages 1975–1980. IEEE, 2007. (Cité en pages 29, 30 et 31.)

[Katz 2007] S. Katz, A. Tal et R. Basri. *Direct Visibility of Point Sets.* ACM Trans. Graph., vol. 26, no. 3, page 24, Juillet 2007. (Cité en page 62.)

[Leclet-Groux 2013] Dominique Leclet-Groux, Guillaume Caron, El Mustapha Mouaddib et Azziz Anghour. *A Serious Game for 3D Cultural Heritage.* pages 409–412, 2013. (Cité en page 63.)

[Lowe 2004] David G. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints.* Int. J. Comput. Vision, vol. 60, no. 2, pages 91–110, Novembre 2004. (Cité en page 28.)

[Marchand 2002a] E. Marchand et F. Chaumette. *Virtual visual servoing : A framework for real-time augmented reality.* In G. Drettakis et H.-P. Seidel, editeurs, EUROGRAPHICS 2002 Conference Proceeding, volume 21(3) of *Computer Graphics Forum*, pages 289–298, SaarebrÃ$\frac{1}{4}$cken, Germany, September 2002. (Cité en page 34.)

[Marchand 2002b] Eric Marchand et Francois Chaumette. *Virtual Visual Servoing : a framework for real-time augmented reality.* Computer Graphics Forum, 2002. (Cité en page 33.)

[Marchand 2005] E. Marchand, F. Spindler et F. Chaumette. *ViSP for visual servoing : a generic software platform with a wide class of robot control skills.* IEEE Robotics and Automation Magazine, vol. 12, no. 4, pages 40–52, December 2005. (Cité en page 87.)

[Marchand 2007] Eric Marchand et François Chaumette. *Fitting 3d models on central catadioptric images.* In IEEE Int. Conf. on Robotics and Automation, ICRA'07, pages 52–58, Roma, Italy, Italy, 2007. (Cité en page 35.)

[Mastin 2009] A. Mastin, J. Kepner et J. Fisher. *Automatic registration of LIDAR and optical images of urban scenes.* IEEE Conference on Compu-

ter Vision and Pattern Recognition, pages 2639–2646, 2009. (Cité en page 60.)

[Morel 2009] Jean-Michel Morel et Guoshen Yu. *ASIFT : A New Framework for Fully Affine Invariant Image Comparison*. SIAM J. Img. Sci., vol. 2, no. 2, pages 438–469, Avril 2009. (Cité en page 61.)

[Moussa 2012] W. Moussa, M. Abdel-Wahab et D. Fritsch. *An automatic procedure for combining digital images and laser scanner data*. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, pages 229–234, 2012. (Cité en page 60.)

[Nayar 1996] S.K. Nayar, S.A. Nene et H. Murase. *Subspace Methods for Robot Vision*. IEEE Transactions on Robotics and Automation, vol. 12, no. 5, pages 750–758, Oct 1996. (Cité en page 24.)

[Shannon 1948] Claude Shannon. *A Mathematical Theory of Communication*. Bell System Technical Journal, vol. 27, pages 379–423, 623–656, 1948. (Cité en page 27.)

[Sundareswaran 1998] R. Sundareswaran S. et Behringer. *Visual servoing based augmented reality*. November 1998. (Cité en page 33.)

[Tahri 2010] O. Tahri, Y. Mezouar, F. Chaumette et H. Araujo. *Visual servoing and pose estimation with cameras obeying the unified model*. In G. Chesi et K. Hashimoto, editeurs, Visual Servoing via Advanced Numerical Methods, pages 231–252. LNCIS 401, Springer-Verlag, 2010. (Cité en page 21.)

[Tahri 2015] O. Tahri, A. Yeremou Tamtsia, Y. Mezouar et Demonceaux C. *Visual Servoing based on Shifted Moments*. Accepted in IEEE Transactions on RObotics, vol. 31, no. 3, pages 798–804, June 2015. (Cité en page 20.)

[Tian 2002] Gui Yun Tian, Duke Gledhill, David Taylor et David Clarke. *Colour Correction for Panoramic Imaging*. Proc. 6th Int. Conf. Inf. Vis., pages 483–488, 2002. (Cité en page 46.)

[Viola 1997] Paul Viola et William M. Wells III. *Alignment by Maximization of Mutual Information*. Int. J. Comput. Vision, vol. 24, no. 2, pages 137–154, Septembre 1997. (Cité en page 27.)

[Wang 2008] Junping Wang et Hyungsuck Cho. *Micropeg and Hole Alignment Using Image Moments Based Visual Servoing Method*. IEEE Transactions on Industrial Electronics, vol. 55, no. 3, pages 1286–1294, 2008. (Cité en page 31.)

[Weiss 1987] Lee Weiss, Arthur C Sanderson et C. P. Neuman. *Dynamic Sensor-Based Control of Robots with Visual Feedback*. IEEE Journal on Robotics and Automation, vol. RA-3, no. 5, October 1987. (Cité en page 19.)

[Wilson 1996] William J. Wilson, Carol C. Williams Hulls et G. S. Bell. *Relative end-effector control using Cartesian position based visual servoing.* IEEE T. Robotics and Automation, vol. 12, no. 5, pages 684–696, 1996. (Cité en page 22.)

[Ying 2004] Xianghua Ying et Zhanyi Hu. *Can We Consider Central Catadioptric Cameras and Fisheye Cameras within a Unified Imaging Model.* In ECCV (1), volume 3021, pages 442–455. Springer, 2004. (Cité en page 11.)

Résumé : Cette thèse aborde l'estimation de pose et le positionnement de caméra sous le même formalisme de l'asservissement visuel. Cette approche est une méthode de contrôle en boucle fermée des mouvements d'un système dynamique en utilisant des données visuelles comme retour d'informations. Les données visuelles peuvent, par exemple, être acquises à partir d'une caméra numérique directement mise en mouvement par le système.

Il est essentiel d'établir une relation entre des caractéristiques visuelles provenant des images de la caméra avec ses déplacements dans l'environnement. Généralement, les caractéristiques visuelles utilisées sont géométriques (points, lignes, etc.). Cependant, des étapes de traitements des images sont nécessaires pour extraire, suivre ou encore segmenter ce type de caractéristiques. Bien qu'ayant été, et étant toujours, très étudiées, ces étapes restent très délicates. À tel point que les résultats des asservissements visuels basés sur ce type de caractéristiques dépendent principalement de la qualité d'extraction de ces mesures dans les images. C'est pourquoi, il a été proposé d'exploiter directement l'apparence de la scène plutôt que des mesures éparses censées la représenter.

Les travaux de l'état-de-l'art proposent d'étendre ce formalisme au calcul de pose de caméra exploitant certains types d'environnements 3D et texturés. Il est alors question d'asservissement visuel virtuel. Ces dernières années, les technologies de numérisation 3D sont devenues très efficaces et permettent aujourd'hui de mesurer rapidement et avec précision la structure spatiale et l'aspect d'un environnement réel. Aucun des travaux d'asservissement visuel virtuel passés n'exploite directement les intensités de l'image, ni d'environnement 3D sous forme de nuage de points colorés issus de ces nouveaux outils de numérisation 3D. C'est l'objet d'une des contributions de cette thèse : l'asservissement visuel virtuel photométrique est défini en vision perspective pour recaler des images numériques quelconques sur des nuages de points 3D, afin d'en améliorer la qualité des couleurs. Puis, l'approche a été étendue avec succès à la vision omnidirectionnelle pour la localisation de robot mobile.

La seconde, et majeure, contribution de la thèse, généralise l'asservissement visuel photométrique en introduisant une nouvelle représentation de l'image par mélange de gaussiennes photométriques. Sa déclinaison au positionnement de robot industriel et au recalage de photos sur nuages de points 3D montre une précision tout aussi excellente que l'asservissement visuel photométrique, en agrandissant considérablement son domaine de convergence.

Mots clés : Asservissements visuels denses, estimation de pose de caméra, mélanges de gaussiennes photométriques, robotique, vision par ordinateur

Contributions to dense visual servoing :
new image representation adapted to virtual and real environments

Abstract : This thesis deals with estimation of camera pose under the visual servoing framework. Visual servoing is a method of closed-loop control of the movements of a dynamic system using visual data as feedback. These visual data are usually obtained using a digital camera displaced in the environment by the system.

It is essential to establish a relationship between visual features obtained in the camera images with its movements in the environment. The commonly used visual features are geometric measures in the images (points, lines, etc.). However, image processings are required for the extraction, the matching, the tracking or the segmentation of these kind of features. Even if these image processings have been, and still are, very studied, they still are challenging operations. To such an extent that the results and the behaviour of visual servoing is highly related to the quality of these image measures. That is why, it has been proposed to directly exploit the scene appearance like the intensity of every pixel in the images. The use of the complete images provides redundancy of information and avoids the need of image processings.

If a digital synthesized model of the environment is known, the camera can be virtualized. In recent years, 3D laser scanners have been more and more improved and have now digital cameras included which assign a color to each acquired points. It is now possible to rapidly extract a geometric and visual representation of an environment. In the state of the art of virtual and visual servoing, it has been proposed to exploit textured 3D models for camera pose estimation. However, none of them directly uses the image luminance or an environment composed by a colored 3D point cloud. This is one of the contributions of the thesis : the photometric virtual and visual servoing is defined to the perspective vision in order to align digital images over the 3D point cloud for improve its visual quality. This approach has been successfully extended to the omnidirectional vision for mobile robot localization.

The second, and substantial, contribution of the thesis generalizes the photometric virtual and visual servoing with a new image representation : the photometric Gaussians mixtures. The use of this new image representation for industrial robot positioning and in order to align digital images over the 3D point cloud has the same accuracy as the photometric visual servoing but significantly enlarges its convergence domain.

Keywords : Dense visual servoing, camera pose estimation, photometric Gaussians mixtures, Robotics, Computer Vision