



HAL
open science

Vers l'assistance à l'exploration pertinente et réaliste d'environnements 3D très denses

Zaynab Habibi

► **To cite this version:**

Zaynab Habibi. Vers l'assistance à l'exploration pertinente et réaliste d'environnements 3D très denses. Autre. Université de Picardie Jules Verne, 2015. Français. NNT : 2015AMIE0028 . tel-03653271

HAL Id: tel-03653271

<https://theses.hal.science/tel-03653271>

Submitted on 27 Apr 2022

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

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



Thèse de Doctorat

*Mention Sciences pour l'Ingénieur
Spécialité Vision par Ordinateur*

présentée à

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

de l'Université de Picardie Jules Verne

par

Zaynab Habibi

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

***Vers l'assistance à l'exploration pertinente et réaliste
d'environnements 3D très denses***

Soutenue le 08/12/2015, après avis des rapporteurs, devant le jury d'examen :

Mme Marie-Odile Berger, Directrice de recherche	Rapporteur
M. Eric Marchand, Professeur	Rapporteur
M. Christophe Chaillou, Professeur	Président
M. Marc Christie, Maître de conférences	Examineur
Mme. Dominique Groux, Maître de conférences HDR	Examineur
M. El Mustapha MOUADDIB, Professeur	Examineur
M. Guillaume CARON, Maître de conférences	Examineur

Remerciements

Au terme de cette thèse, je tiens à remercier toutes les personnes ayant contribué de près ou de loin à la réalisation de ce travail.

Je tiens tout d'abord à adresser mes remerciements les plus sincères à El Mustapha Mouaddib pour avoir dirigé cette thèse et m'avoir permis de la réaliser dans les meilleures conditions. Je tiens particulièrement à le remercier de son encadrement scientifique, son suivi régulier, sa patience et ses conseils multiples et pertinents. Je lui exprime aussi ma gratitude pour sa supervision et ses relectures lors de la rédaction du manuscrit, d'articles ainsi que dans la préparation de la soutenance. Enfin, j'ai été extrêmement sensible à ses grandes qualités humaines d'écoute et de compréhension ainsi qu'à son sourire et à sa bonne humeur dont il ne s'est jamais départie tout au long de ces trois années de thèse.

Mes remerciements vont aussi à mon encadrant Guillaume Caron pour son enthousiasme permanent, sa disponibilité et pour toute l'aide qu'il m'a apporté, que ce soit sur les aspects scientifiques ou techniques. Sa relecture très méticuleuse de chacun des chapitres ainsi que des articles m'a sans aucun doute permis d'améliorer la qualité de présenter mon propos et de le préciser. Je lui suis également très reconnaissante du soutien et de toute l'aide pédagogique qu'il m'a apporté lors de mes premiers pas dans l'enseignement supérieur.

Je tiens aussi à remercier mon jury de thèse, à commencer par Christophe Chaillou qui m'a fait l'honneur de le présider. Mes remerciements s'adressent aussi à mes rapporteurs Marie-Odile Berger et Eric Marchand, pour avoir accepté de lire et d'évaluer le manuscrit et pour les commentaires qu'ils ont fait sur mon travail. Je remercie aussi les deux examinateurs Dominique Groux et Marc Christie pour les questions et les idées intéressantes proposées lors de la soutenance.

Je pense également à Nathan Crombez ami doctorant du laboratoire qui travaille aussi comme moi dans le cadre du programme de recherche E-cathédrale pour tous les échanges scientifiques qu'on a pu avoir et aussi pour son aide dans de multiples domaines.

Je n'oublie pas bien-sûr tous les autres amis doctorants et postdocs du laboratoire, et plus particulièrement à Nicolas, Doha, Nour eddine, Clement, Youssef,

Hanane et Fatima Zahra de les remercier pour tous les moments qu'on a pu passer ensemble et pour la bonne ambiance lors des pauses café. Je garderai d'excellents souvenirs de toutes ces années passées avec eux.

Enfin, j'aimerais remercier mes parents, mon frère ainsi que toute ma famille pour leur soutien, surtout moral dans les périodes difficiles que j'ai pu avoir durant la préparation de la thèse.

Table des matières

Introduction	1
1 Outils de base	7
1.1 Introduction	7
1.2 Principes en vision par ordinateur	8
1.2.1 Géométrie projective et matrice de transformation	8
1.2.2 Projection perspective dans le plan image	9
1.3 Asservissement visuel	12
1.3.1 Optimisation non-linéaire	12
1.3.2 Classification des méthodes d'asservissement visuel	14
1.4 Conclusion	23
2 Assistance à la navigation : outils et méthodes	25
2.1 Introduction	25
2.2 Pertinence d'un point de vue	26
2.2.1 Méthodes photométriques	27
2.2.2 Méthodes géométriques	28
2.3 Contrôle de caméra virtuelle	34
2.3.1 Méthodes manuelles et assistées	35
2.3.2 Méthodes automatiques	38
2.4 Conclusion	42
3 Cadrage visuel	44
3.1 Introduction	45
3.2 Cadrage visuel automatique basé entropie photométrique	45
3.2.1 Définition statistique de l'entropie de Shannon	46
3.2.2 Modélisation du problème : Entropie photométrique	46
3.2.3 Résolution du problème	48
3.2.4 Résultat sur un nuage de points 3D dense et coloré	51
3.2.5 Limitations	52
3.3 Cadrage visuel automatique : mélange de gaussiennes basées saillance	55
3.3.1 Critère de la saillance visuelle	55

3.3.2	Le mélange de gaussiennes représentant l'image de saillance	58
3.3.3	Définition des paramètres des gaussiennes	59
3.3.4	Expression optimale du λ	60
3.3.5	Résolution du problème	62
3.3.6	Résultat de simulation : environnement 2D	64
3.3.7	Résultat de simulation : environnement 3D	65
3.3.8	Résultat sur un nuage de points dense et coloré	68
3.3.9	Résultat en environnement réel	73
3.4	Cadrage visuel assisté	76
3.4.1	Loi de commande assistée	77
3.4.2	Résultat de simulation	79
3.4.3	Résultat sur un nuage de points 3D	79
3.4.4	Évaluation du système	82
3.5	Conclusion	85
4	Exploration virtuelle d'un environnement 3D	88
4.1	Exploration automatique basée entropie visuelle	89
4.1.1	Formalisation du problème : tâche et contraintes	89
4.1.2	Schéma de résolution : hybride et hiérarchique	92
4.1.3	Résultats expérimentaux	97
4.1.4	Conclusion	101
4.2	Exploration assistée : mélanges de gaussiennes basées saillance	102
4.2.1	Formalisation sous forme de contraintes	104
4.2.2	Schéma de résolution : redondance et hiérarchie	107
4.2.3	Résultats expérimentaux	111
4.2.4	Conclusion	114
	Conclusion et perspectives	119
	Bibliographie	123

Table des figures

1	Images de <i>E-Cathédrale</i>	2
2	(a) FARO focus 3D, (b) Leica C10.	3
3	Modèle virtuel de la cathédrale d'Amiens.	3
1.1	Représentation du repère objet et caméra.	9
1.2	Projection perspective.	10
1.3	Représentation de la conversion de l'espace métrique à l'espace pixelique.	11
1.4	Séquence d'images relative au cadrage d'un tableau [Courty 2002].	21
1.5	Positionnement de la caméra par rapport à une sphère en maximisant la luminance [Marchand 2007].	23
2.1	(a) off-center des cellules ganglionnaires, (b) on-center des cellules ganglionnaires.	28
2.2	Cartes de saillance calculées en utilisant [Radhakrishna 2009].	29
2.3	L'entropie d'un point de vue (illustration de la surface projetée sur la sphère) [Vázquez 2001].	30
2.4	Résultat du calcul de la saillance sur un maillage (les zones vertes et jaunes sont les plus saillantes) [Lee 2005].	31
2.5	Résultat du calcul de la saillance sur un nuage de points dense (les zones jaunes, oranges et rouges sont les plus saillantes) [Shtrom 2013].	32
2.6	(a)Résultat obtenu avec notre implémentation, (b) Résultat obtenu dans [Shtrom 2013].	33
2.7	(a)Résultat obtenu avec notre implémentation, (b) Résultat obtenu dans [Shtrom 2013].	34
2.8	(a)Extrait du nuage intérieur de la cathédrale, (b) Saillance 3D : point de vue 1, (c) Saillance 3D : point de vue 2.	35
2.9	Quelques types de mouvements de caméra.	36
2.10	Copie écran du jeu Wolfenstein enemy territory (métaphore " <i>œil en main</i> ").	36
2.11	Les Types de mouvement dans <i>Hovercam</i> [Khan 2005].	37
2.12	Illustration de <i>Navidget</i> basé sur la technique des positions d'intérêts sur plusieurs systèmes d'affichage [Hachet 2008].	38
2.13	Exploration d'une ville à l'aide de la méthode [Hagedorn 2008], l'utilisateur dessine la trajectoire que doit suivre la caméra.	39

2.14 Déplacement de l'avatar à l'aide du calcul de la "feuille de route" [T. Li 2008].	40
2.15 Génération automatique d'une trajectoire passant par toutes les cellules [Andújar 2004].	40
2.16 Cadrage du tableau en se basant sur des primitives de l'image [Courty 2001].	41
3.1 La trajectoire visuelle de la caméra en maximisant la quantité d'information pertinente dans l'image.	52
3.2 Évolution de l'entropie photométrique.	52
3.3 Résultat de cadrage en maximisant l'entropie photométrique.	53
3.4 Un échantillon de la séquence d'images obtenue en maximisant l'entropie photométrique.	54
3.5 (a) L'évolution de l'entropie photométrique; (b) Les vitesses des 4 d.d.l de la caméra.	55
3.6 Le signal avec une seule valeur saillante. La fonction de coût reste constante quelque soit la translation du signal dans le cadre.	56
3.7 La fonction gaussienne pour un signal avec une seule valeur de saillance.	57
3.8 La fonction du mélange de gaussiennes pour un signal constitué de trois valeurs saillantes.	58
3.9 (a) Image de saillance, (b) l'image sous forme d'impulsions.	59
3.10 Influence du λ sur le mélange de gaussiennes considérant une image simple avec un seul pixel saillant.	60
3.11 (a) Fonction Gaussienne avec $\sigma = 1$, (b) première dérivée de la gaussienne, (c) seconde dérivée de la gaussienne.	61
3.12 Le mélange de gaussiennes associé à l'image en utilisant la méthode proposée pour le calcul du λ	62
3.13 Cadrage visuel : mélange de gaussiennes basées saillance.	64
3.14 Cas extrême de cadrage en utilisant la primitive mélange de gaussiennes basées saillance.	65
3.15 loi de commande mélange de gaussiennes basées saillance dans un environnement 3D simple.	66
3.16 Navigation automatique sans rotation en utilisant la primitive mélange de gaussiennes basées saillance.	68
3.17 Navigation automatique avec rotation en utilisant la primitive mélange de gaussiennes basées saillance.	69
3.18 Impact de la résolution de l'image sur le mélange de gaussiennes.	70

3.19	Projection cylindrique.	71
3.20	L'environnement expérimental 3D du portail sud de la cathédrale. .	73
3.21	Loi de commande hybride.	74
3.22	loi de commande hybride.	75
3.23	Le robot utilisé dans nos expériences.	76
3.24	Navigation automatique du robot.	77
3.25	Différentes positions atteintes par le robot durant le cadrage visuel.	77
3.26	Navigation automatique du robot.	78
3.27	Cadrage visuel assisté dans un environnement de synthèse.	80
3.28	Exemple 1 : Séquence d'images pour le cadrage assistée.	81
3.29	Exemple 1 : Trajectoire de la caméra pour le cadrage assistée	82
3.30	Exemple 2 : Séquence d'images pour le cadrage assistée.	83
3.31	Exemple 2 : Trajectoire de la caméra pour le cadrage assistée. . . .	83
3.32	Evaluation de l'utilisateur des méthodes de cadrage.	84
4.1	La stratégie hiérarchique.	91
4.2	Exemple de régularisation de l'amplitude du mouvement dans l'image de deux pixels A et B.	92
4.3	Processus de contrôle de caméra.	94
4.4	Calcul du flot optique théorique (calcul de la différence entre les coordonnées des points des images rendues à la pose \mathbf{r}_{k-1} et $\mathbf{r}_{k-1} + \dot{\mathbf{r}}_k$). .	95
4.5	Navigation automatique en maximisant l'entropie photométrique : vue d'extérieur du modèle.	98
4.6	Évolution de l'entropie photométrique. les points bleus correspondent aux images de la Figure 4.5 (de gauche à droite et de haut en bas). .	98
4.7	Comportement de la contrainte d'évitement d'obstacles.	99
4.8	Trajectoire de la caméra.	100
4.9	Comparaison flot optique moyen.	101
4.10	Navigation automatique en extérieur.	101
4.11	Deuxième exemple : séquences d'image de navigation intérieure. . .	102
4.12	Deuxième exemple : séquence d'images de navigation intérieure. . .	103
4.13	Illustration de l'exploration d'un musée.	105
4.14	Contenu similaire.	109
4.15	Trajectoire : exploration assistée (simulation)	112
4.16	Résultat en simulation de l'exploration assistée.	113
4.17	(a) Un échantillon des images rendues en utilisant la primitive pho- tométrique pour le contrôle de l'orientation, (b) Un échantillon des images rendues en utilisant l'asservissement visuel basé poses. . . .	114

4.18	Séquence d'images pour la navigation assistée.	115
4.19	Les positions cibles sont en bleu et la trajectoire de la caméra en rouge.	115
4.20	Trajectoire de la caméra en respectant la contrainte de l'évitement des obstacles.	116
4.21	Trajectoire de la caméra sans respecter la contrainte de l'évitement des obstacles.	116

Introduction

Le patrimoine culturel est un héritage du passé et constitue une référence à notre histoire et notre identité. Sa protection et sa préservation représente un enjeu collectif, d'où la nécessité d'employer les moyens humains et techniques pour l'assurer. Avec l'appui des technologies de numérisation, la préservation du patrimoine est facilitée dans une perspective à long terme. La numérisation offre l'opportunité de modéliser avec la plus haute fidélité et précision le monde réel. Son utilisation dans le cadre de la préservation du patrimoine permet de le rendre accessible à différents publics à l'instant présent et dans le futur. La complexité et la taille des édifices architecturaux sujets à la numérisation implique l'utilisation d'ordinateurs dotés d'une puissance de calcul graphique très importante, ce qui a toujours représenté un goulot d'étranglement dans ce domaine de recherche. Les environnements virtuels 3D obtenus à l'aide de la numérisation donnent la possibilité de développer plusieurs applications en rapport avec la réalité virtuelle par l'intermédiaire de la navigation virtuelle et des jeux.

Ces environnements virtuels sont exploités dans cette thèse qui s'inscrit dans le cadre du projet de recherche *ASSIDUITAS* soutenu par la région Picardie et le FEDER. Ce travail consiste à réaliser un système d'aide à la navigation visuelle pertinente dans le modèle 3D de la cathédrale gothique Notre Dame d'Amiens. Inscrite par l'UNESCO sur la liste du patrimoine mondial, elle a été construite entre 1220 et 1270. Elle mesure 145 *m* en longueur, 70 *m* en largeur et 42,30 *m* de hauteur sous la voûte.

Plusieurs campagnes de numérisation ont été effectuées dans le cadre de *E-Cathédrale*, qui est un programme de recherche et de développement d'une maquette 3D de la cathédrale d'Amiens (www.mis.u-picardie.fr/E-Cathedrale/). Ce programme vise à numériser sur une période de quinze ans complètement et précisément la cathédrale d'Amiens (Figure 1). Des scanners lasers à balayage *Leica C10* et *FARO Focus 3D* (Figure 2) sont utilisés pour l'acquisition qui opère selon une succession de deux phases. Dans la première, le scanner laser mesure précisément la structure 3D du monument et génère ainsi un ensemble de points 3D. Ensuite, une caméra accompagnant le scanner laser permet de prendre des images dans un champ de vue de 360 degrés afin d'attribuer les couleurs des images aux points 3D. Compte tenu de la taille de la cathédrale d'Amiens, cette étape doit être réalisée à de multiples reprises en déplaçant le scanner laser selon plusieurs stations. La

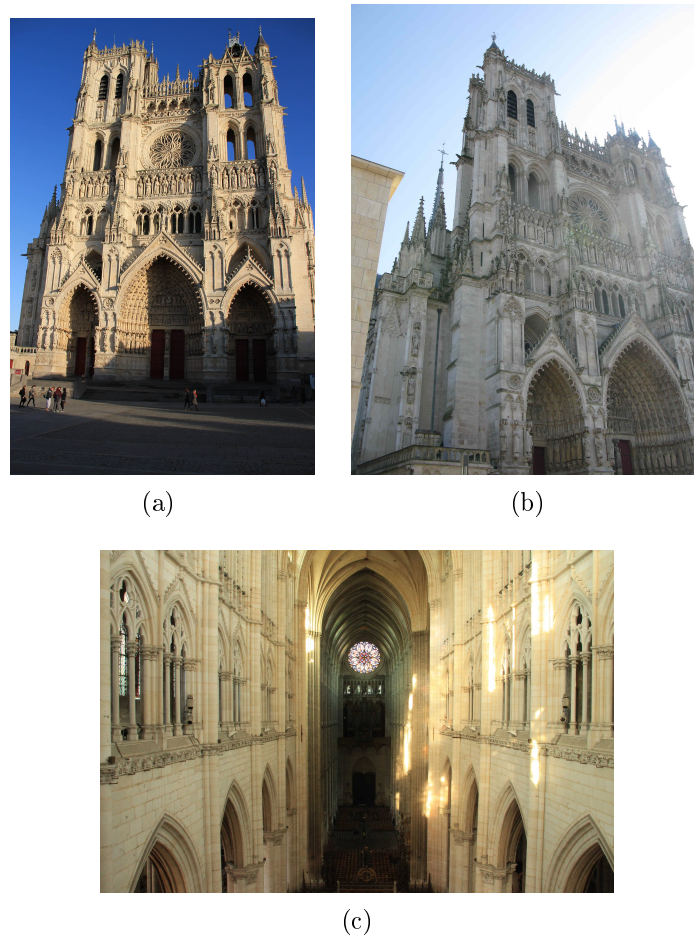


FIGURE 1 – Images de *E-Cathédrale*.

Figure 3 montre une vue du nuage de points de 3D de la cathédrale obtenu à l'aide de ces scanners laser.

La visualisation et la navigation dans un environnement virtuel constitué d'un vaste et complexe nuage de points 3D est un véritable défi dans la mesure où il est difficile de bien s'orienter ou de se situer dans le modèle. Parmi les logiciels existants pour la visite des environnements virtuels, il y a ceux qui ne permettent aucune interaction et ceux qui attribuent à l'utilisateur une liberté totale de mouvement selon tous les degrés de liberté de la caméra (d.d.l), les trois translations et les trois rotations ce qui conduit à une navigation irréaliste (la désorientation) et non pertinente (visualisation de zones pauvres en information). C'est la raison pour laquelle les techniques d'assistance à l'interaction ont vu le jour. Elles s'exécutent en arrière plan afin de faciliter l'interaction avec le modèle.



FIGURE 2 – (a) FARO focus 3D, (b) Leica C10.

Durant ces dernières années, la navigation virtuelle a retenu l'attention des chercheurs dans le domaine de la réalité virtuelle et de la vision par ordinateur. Cependant, rares sont les travaux qui utilisent l'information visuelle, extraites de l'image pour guider la caméra durant son exploration. Dans le cas des modèles constitués d'un nuage de points 3D, ce sont les méthodes de planification de trajectoire qui sont souvent utilisées.

L'objectif de cette thèse est d'utiliser les techniques de vision par ordinateur pour



FIGURE 3 – Modèle virtuel de la cathédrale d'Amiens.

réaliser la navigation virtuelle par contrôle de caméra. Nous proposons d'abord une méthode de contrôle automatique de la caméra. Ensuite, cette technique est étendue à un système de navigation virtuelle assistée qui combine les entrées utilisateur au contrôle automatique afin de l'aider dans sa tâche d'exploration. Ce système est divisé en deux parties, qui sont le cadrage visuel assisté destiné à un contrôle de caméra local et une visite virtuelle qui constitue une stratégie de navigation plus globale. La technique de contrôle de caméra employée vise à utiliser des informations 2D, sous forme de primitive visuelle (l'image rendue) et des informations 3D (les points 3D) pour commander la caméra virtuelle. La caméra est guidée de manière à garantir la pertinence visuelle qui a fait l'objet d'une étude poussée. Elle est formulée de deux manières différentes dans cette thèse. La première s'appuie sur l'hypothèse qu'un utilisateur est essentiellement attiré par les zones riches en information. Ceci est modélisé à l'aide de la théorie de l'information et plus précisément de l'entropie de Shannon. Elle est calculée en utilisant l'information photométrique de l'image. Ensuite, partant du constat, fait par la théorie de la perception visuelle, et qui souligne l'importance que joue la saillance dans l'exploration visuelle, nous l'avons retenue comme critère important de la pertinence visuelle. Ainsi, la saillance dans l'image va être utilisée comme primitive visuelle. Mais l'utilisation directe a des limites qui seront explicitées au chapitre 3. Une nouvelle modélisation basée sur les mélanges de gaussiennes a été proposée. Le lien entre le déplacement de la caméra selon ses d.d.l et les fonctions de pertinence visuelle est assuré grâce au formalisme de l'asservissement visuel. Il permet de mettre à jour d'une manière itérative les d.d.l de la caméra en optimisant ces critères basés image. Par la suite, afin de garantir un mouvement réaliste, quelques contraintes ont été introduites afin d'assurer l'évitement d'obstacles, la fluidité du mouvement, une orientation appropriée de la caméra et un contenu visuel non similaire. Enfin, la combinaison de la contrainte principale relative à la pertinence visuelle et de toutes les autres est réalisée à l'aide d'une loi de contrôle hybride et hiérarchique.

Ce mémoire est organisé selon quatre chapitres :

- Le premier chapitre présente quelques principes en vision par ordinateur exploités dans les méthodes que nous proposons. Ensuite, une étude sur les différents types d'asservissement visuel est établie.
- Le deuxième chapitre compare les méthodes de contrôle de caméra, mettant en avant leurs avantages et leurs limitations. Ensuite, une étude sur la définition de la pertinence visuelle dans la littérature est détaillée.
- Le troisième chapitre concerne la méthode de cadrage automatique et assis-

tée proposée. Une comparaison avec différentes fonctions de pertinence est réalisée. Une nouvelle primitive visuelle est introduite, qui est un mélange de gaussiennes basées saillance. Différentes expérimentations, en simulation, sur un nuage de points 3D et en environnement réel ont été menées pour la validation de l'approche proposée.

- Le quatrième chapitre traite deux méthodes d'exploration d'un environnement complexe et de taille conséquente constitué d'un nuage de points 3D. La première méthode exploite l'entropie photométrique comme fonction de pertinence accompagnée de trois contraintes : l'évitement d'obstacles, le réalisme des rotations et la fluidité du mouvement. Toutes ces contraintes sont combinées et un schéma de résolution hybride et hiérarchique est employé pour le calcul de la loi de contrôle. Dans la deuxième méthode, l'utilisateur entre en jeu en définissant les zones qu'il souhaite visiter du modèle à l'aide de positions de la caméra. Le déplacement d'une position à l'autre est réalisé à l'aide d'un asservissement visuel basé pose. A ce dernier, s'ajoutent différentes contraintes pour assurer une navigation réaliste et pertinente. Le schéma de résolution s'appuie sur une loi de contrôle hiérarchique qui utilise le formalisme de redondance.
- Enfin, ce mémoire s'achève par une conclusion et quelques perspectives de recherche envisageables.

Outils de base

Sommaire

1.1	Introduction	7
1.2	Principes en vision par ordinateur	8
1.2.1	Géométrie projective et matrice de transformation	8
1.2.2	Projection perspective dans le plan image	9
1.3	Asservissement visuel	12
1.3.1	Optimisation non-linéaire	12
1.3.2	Classification des méthodes d'asservissement visuel	14
1.4	Conclusion	23

1.1 Introduction

L'assistance à la navigation passe par le pilotage de la caméra virtuelle. Notre approche consiste à effectuer ce pilotage en fonction d'informations visuelles. Il est donc nécessaire de modéliser la relation entre l'information 2D issue de l'image et ce à quoi elle correspond dans le monde 3D, qu'il soit réel ou virtuel. Ce lien est modélisé à l'aide de transformations. Le calcul de ces transformations repose sur l'exploitation des informations 2D perçues dans l'image acquise par cette caméra. Dans ce chapitre, nous allons d'abord présenter le modèle de projection perspective qui consiste en une représentation 2D (une image) du monde 3D. Nous nous sommes restreints à ce modèle, compte tenu de son utilisation tout au long des travaux de cette thèse. Ensuite, nous allons nous intéresser aux techniques d'asservissement visuel qui consistent à utiliser l'information 2D extraite des images, ou 3D afin de positionner et d'orienter la caméra par rapport au repère objet.

1.2 Principes en vision par ordinateur

1.2.1 Géométrie projective et matrice de transformation

Un point \mathbf{X} dans l'espace euclidien peut être représenté par ses coordonnées homogènes $(X, Y, Z, 1)$ dans l'espace projectif tel que (X, Y, Z) sont les coordonnées cartésiennes qui appartiennent à \mathbb{R}^3 .

On considère le point ${}^{R_1}\mathbf{X}$ dans \mathcal{R}_1 et le point ${}^{R_2}\mathbf{X}$ dans \mathcal{R}_2 , tel que \mathcal{R}_2 et \mathcal{R}_1 sont deux repères orthonormés centrés en \mathbf{O}_1 et \mathbf{O}_2 , respectivement. La transformation 3D qui lie ${}^{R_1}\mathbf{X}$ et ${}^{R_2}\mathbf{X}$ est définie par la matrice ${}^{R_2}\mathbf{M}_{R_1}$ tel que :

$${}^{R_2}\mathbf{X} = {}^{R_2}\mathbf{M}_{R_1} {}^{R_1}\mathbf{X} \quad (1.1)$$

Dans le cas général, le passage du repère \mathcal{R}_1 au repère \mathcal{R}_2 est une combinaison de translations et de rotations :

$${}^{R_2}\mathbf{X} = \begin{pmatrix} {}^{R_2}\mathbf{R}_{R_1} & {}^{R_2}\mathbf{t}_{R_1} \\ 0 & 1 \end{pmatrix} {}^{R_1}\mathbf{X} \quad (1.2)$$

Tel que, ${}^{R_2}\mathbf{t}_{R_1}$ est un vecteur de translation ${}^{R_2}\mathbf{t}_{R_1} = (t_X, t_Y, t_Z) \in \mathbb{R}^3$ et ${}^{R_2}\mathbf{R}_{R_1}$ est une matrice 3×3 du groupe $\mathcal{SO}(3)$.

$\mathcal{SO}(3)$ est le groupe spécial orthogonal défini par :

$$\mathcal{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3 \times 3} \mid \mathbf{R}^\top \mathbf{R} = \mathbf{I}_3 \text{ et } \det(\mathbf{R}) = \pm 1\} \quad (1.3)$$

Ainsi, la matrice de transformation entre le repère objet \mathcal{R}_o et le repère caméra \mathcal{R}_c (Figure 1.1) est définie par :

$${}^c\mathbf{M}_o = \begin{pmatrix} {}^c\mathbf{R}_o & {}^c\mathbf{t}_o \\ 0 & 1 \end{pmatrix} \quad (1.4)$$

Cette dernière matrice, définissant la transformation effectuée, représente les paramètres extrinsèques de la caméra. Elle est constituée de la position et l'orientation par rapport au repère objet.

Dans cette thèse, nous avons choisi d'exprimer les rotations à l'aide de la représentation exponentielle canonique $\theta\mathbf{w} = \{\theta w_x, \theta w_y, \theta w_z\}$, qui est constituée d'un vecteur unitaire de rotation $\mathbf{w} = \{w_x, w_y, w_z\}$ et d'un angle de rotation θ . Ainsi, d'après la formule de *Rodriguès* la matrice de rotation \mathbf{R} est calculée en utilisant l'application exponentielle :

$$\mathbf{R} = \exp(\theta[\mathbf{w}]_\times) \quad (1.5)$$

où $[\mathbf{w}]_{\times}$ désigne la matrice antisymétrique de \mathbf{w} :

$$[\mathbf{w}]_{\times} = \begin{pmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{pmatrix} \quad (1.6)$$

Cette représentation peut être confrontée à des singularités pour $\theta = n\Pi$ (tel que $n > 0$) qui sont facilement gérées en calculant une nouvelle matrice de rotation équivalente loin de la singularité.

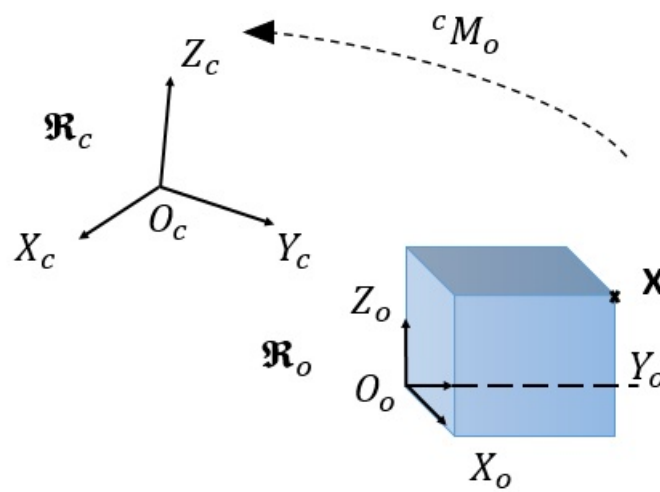


FIGURE 1.1 – Représentation du repère objet et caméra.

1.2.2 Projection perspective dans le plan image

Le modèle sténopé permet le passage d'un modèle 3D (l'objet) à un plan 2D (l'image) à travers un trou infiniment petit. L'image est créée à partir de la propagation rectiligne de la lumière à travers ce trou. Chaque point éclairé appartenant à la surface de l'objet réfléchit les rayons de la lumière dans toutes les directions. Le trou laisse passer les rayons qui rencontrent le plan de projection où ils produisent une image inversée de l'objet.

Dans ce modèle, le trou est représenté par un point appelé centre de projection qui est l'origine du repère Euclidien de la caméra. Un point est projeté en un point dans l'image $\mathbf{x} = (x, y)$ à l'aide de la ligne reliant le point \mathbf{X} et le centre de la caméra \mathbf{O}_c , appelé centre de projection (Figure 1.2a). Le plan image est parallèle

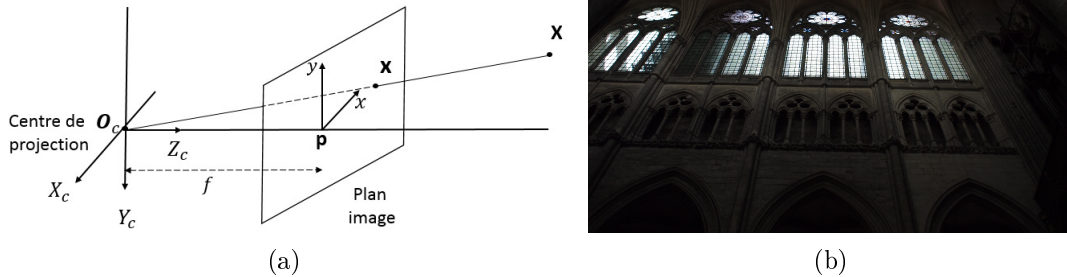


FIGURE 1.2 – (a) Modèle de projection de la caméra "trou d'aiguille". \mathbf{O}_c est le centre de projection et \mathbf{p} le point principal (centre du plan image), (b) Exemple d'une image issue de la projection perspective de l'intérieur de la cathédrale d'Amiens (où les fenêtres ont la même largeur, sur toute leur longueur dans l'espace).

au plan (\vec{X}_c, \vec{Y}_c) et a pour centre le point principal \mathbf{p} . La distance entre le centre de projection et le point principal est la focale f .

Soit ${}^c\mathbf{X}$ le point 3D dans le repère caméra \mathcal{R}_c , le processus permettant de passer à \mathbf{x} dans le repère du plan image métrique est la projection perspective :

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{{}^cZ} \begin{bmatrix} {}^cX \\ {}^cY \end{bmatrix} \quad (1.7)$$

En utilisant les coordonnées homogènes correspondantes $\mathfrak{X} = ({}^cX, {}^cY, {}^cZ, 1)$ et $\mathbf{x} = (wx, wy, w)$ (tel que w est un scalaire), l'équation (1.7) devient :

$$\begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} {}^cX \\ {}^cY \\ {}^cZ \\ 1 \end{bmatrix} \quad (1.8)$$

Une image digitale (Figure 1.3) est représentée sous la forme d'une matrice de valeurs (e.g. niveaux de gris, couleurs RVB) avec pour cellules les pixels. La taille de cette matrice est $h \times l$, tel que l est la largeur en pixels de l'image et h sa hauteur. Un changement de repère doit être envisagé pour utiliser directement les coordonnées 2D définies dans le repère image pixélique.

Les coordonnées normalisées $\mathbf{x} = (x, y, 1)$ sont définies dans le plan image métrique, représentées dans l'espace métrique. Alors que les pixels $\mathbf{u} = (u, v, 1)$ sont

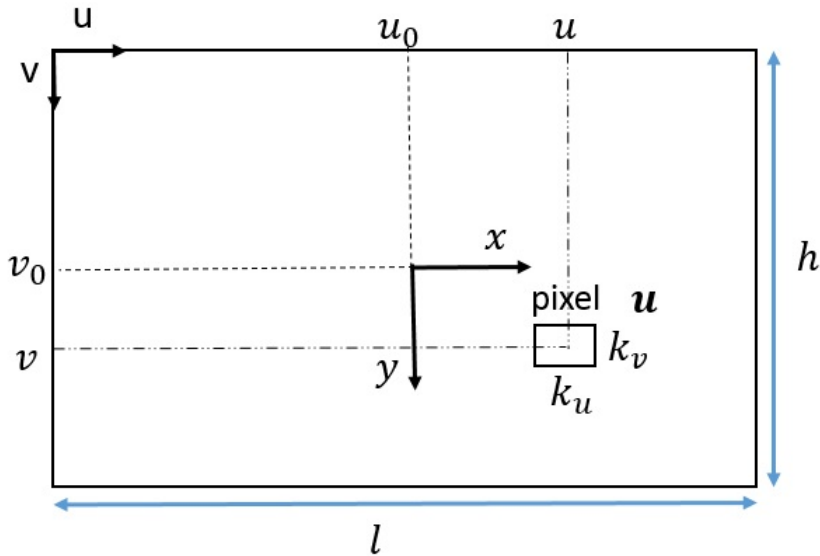


FIGURE 1.3 – Représentation de la conversion de l'espace métrique à l'espace pixelique.

définis dans l'espace pixelique. Le passage des uns aux autres nécessite une transformation de quatre d.d.l. Cette transformation est définie par les coordonnées du point principal (u_0, v_0) et par les paramètres de changement d'échelle (α_u, α_v) . Ces paramètres prennent en compte la focale f en terme de distance et les facteurs d'échelle k_u, k_v qui permettent de convertir cette distance en pixels de telle sorte que :

$$\alpha_u = f k_u \quad (1.9)$$

et,

$$\alpha_v = f k_v \quad (1.10)$$

Ainsi, la transformation pour passer de \mathbf{x} à \mathbf{u} est donnée par \mathbf{K} , appelée matrice des paramètres intrinsèques :

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1.11)$$

Ces paramètres représentent les caractéristiques internes de la caméra (paramètres intrinsèques), qui sont invariants par rapport à sa position. Ils sont généralement fournis par le constructeur ou estimés à l'aide de méthodes de calibrage [Duane 1971] [Zhang 2000] [Caron 2011].

1.3 Asservissement visuel

L'asservissement visuel [Weiss 1987] [Hutchinson 1996] [Chaumette 2006] consiste à contrôler automatiquement, dans une boucle fermée, les d.d.l d'une caméra embarquée sur un robot ou une caméra se déplaçant dans un environnement virtuel. Dans cette thèse, l'un des objectifs principaux est de contrôler automatiquement une caméra en s'appuyant sur les méthodes d'asservissement visuel. En effet, le but est de calculer les poses de la caméra dans un environnement virtuel 3D ou un environnement réel en utilisant des informations 2D ou 3D. La trajectoire réalisée est constituée d'un ensemble de poses (positions et orientations), ainsi le problème est considéré comme un problème d'estimation de pose itératif. Dans la mesure où toutes les fonctions à optimiser qui seront introduites pour réaliser le contrôle de la caméra sont non-linéaires, une résolution en utilisant un processus d'optimisation non-linéaire itératif va être adoptée. Il consiste à estimer un incrément de pose permettant de passer d'une pose courante \mathbf{r}_k à une pose \mathbf{r}_{k+1} en minimisant une fonction de coût ou d'erreur. C'est sur ces principes que l'asservissement visuel se base. Dans cette section, quelques bases sur l'optimisation non-linéaire sont présentées, ensuite une classification de quelques méthodes existantes d'asservissement visuel est réalisée.

1.3.1 Optimisation non-linéaire

A partir d'une pose initiale de la caméra, l'estimation de son déplacement est calculée à l'aide d'une méthode d'optimisation non-linéaire itérative. Le but étant de déplacer la caméra répondant à un ou plusieurs critères formulés à l'aide d'une fonction de coût $f(\mathbf{r})$:

$$\hat{\mathbf{r}} = \underset{\mathbf{r}}{\operatorname{arg\,min}}(f(\mathbf{r})) \quad (1.12)$$

Cette fonction a pour paramètre la pose de la caméra $\mathbf{r} = (t_X, t_Y, t_Z, \theta w_X, \theta w_Y, \theta w_Z)^\top$, où les trois translations sont $[t_X, t_Y, t_Z]^\top$ et la rotation est exprimée à l'aide de sa représentation canonique exponentielle comme vu dans la section 1.2.1. Lorsque $f(\mathbf{r})$ est non linéaire, et c'est notre cas, le schéma itératif permet de résoudre le problème. Il consiste à estimer l'incrément de pose $\dot{\mathbf{r}}_k$, correspondant à la vitesse de la caméra à l'itération k afin de modifier la pose courante \mathbf{r}_k , à l'aide de l'exponentielle map $e^{[\dot{\mathbf{r}}_k]}$:

$$\mathbf{r}_{k+1} = e^{[\dot{\mathbf{r}}_k]} \mathbf{r}_k \quad (1.13)$$

Les méthodes d'optimisation non-linéaire sont basées sur la linéarisation de la fonction de coût à l'aide du développement de Taylor, en une fonction dont les

coefficients dépendent des dérivées d'ordre 1 à n de la fonction de coût au voisinage d'un point.

Soit $f(\mathbf{r})$ une fonction de classe C^n (n fois dérivable) sur $\mathbb{R}^3 \times \mathcal{SO}(3)$. Le développement de Taylor de la fonction $f(\mathbf{r})$ au premier ordre donne :

$$f(\mathbf{r}_{k+1}) \simeq f(\mathbf{r}_k) + \frac{\partial f(\mathbf{r}_k)}{\partial \mathbf{r}} \mathbf{r}_k \dot{\mathbf{r}}_k \quad (1.14)$$

Pour la résolution de ce problème, il y a plusieurs méthodes comme la descente du gradient et celle de Newton. Il y a aussi l'algorithme de Levenberg-Marquardt qui est une combinaison de la méthode de descente du gradient et la méthode de Newton. Il permet une convergence rapide tout en réglant d'une manière itérative le pas de convergence. Toutefois, il ne sera pas utilisé dans cette thèse car nous souhaitons garder le contrôle sur l'amplitude du mouvement tout en jouant sur le pas de convergence, comme on le verra au Chapitre 4.

Descente du gradient : L'algorithme de descente du gradient est le plus simple en optimisation non-linéaire, dont le but est de partir d'une pose initiale et puis de se déplacer dans la direction de la plus forte pente de la fonction de coût. Pour résoudre le problème de minimisation d'une fonction de coût non linéaire qui a pour paramètre le vecteur de pose \mathbf{r} , le vecteur gradient négatif $-\frac{\partial f(\mathbf{r}_0)}{\partial \mathbf{r}}$ qui définit la direction de la plus grande descente est exploité. Ainsi, pour minimiser f , il suffit de se déplacer itérativement dans la direction du gradient. L'amplitude du pas peut être calculée en effectuant une recherche linéaire du minimum en fonction de la direction du gradient négatif :

$$\lambda \dot{\mathbf{r}} = -\frac{\partial f(\mathbf{r}_0)}{\partial \mathbf{r}} \quad (1.15)$$

où λ est un paramètre scalaire qui contrôle la longueur du pas.

La méthode de descente du gradient utilisée toute seule n'est pas une méthode efficace de minimisation car elle est caractérisée par une convergence lente.

Gauss-Newton : Cette méthode permet de trouver le minimum d'une fonction de coût non linéaire qui a pour paramètre le vecteur de pose \mathbf{r} . Le problème d'optimisation consiste simplement à minimiser $f(\mathbf{r})$ par rapport à toutes les valeurs du vecteur \mathbf{r} . On fait l'hypothèse que la fonction f possède un minimum bien défini et que l'on connaît un paramètre \mathbf{r}_0 raisonnablement proche du minimum.

Le développement de Taylor de la fonction f par rapport à \mathbf{r}_0 nous amène à :

$$f(\mathbf{r}_0 + \dot{\mathbf{r}}) \simeq f(\mathbf{r}_0) + \frac{\partial f(\mathbf{r}_0)}{\partial \mathbf{r}} \dot{\mathbf{r}} + \dot{\mathbf{r}}^T \frac{\partial^2 f(\mathbf{r}_0)}{\partial \mathbf{r}^2} \dot{\mathbf{r}} \quad (1.16)$$

On souhaite minimiser cette quantité par rapport à $\dot{\mathbf{r}}$. Pour ce faire, on dérive la fonction de coût par rapport à $\dot{\mathbf{r}}$ et on pose que la dérivée est égale à zéro pour arriver à l'équation suivante :

$$\frac{\partial^2 f(\mathbf{r}_0)}{\partial \mathbf{r}^2} \dot{\mathbf{r}} = - \frac{\partial f(\mathbf{r}_0)}{\partial \mathbf{r}} \quad (1.17)$$

où $\frac{\partial^2 f(\mathbf{r}_0)}{\partial \mathbf{r}^2}$ est la Hessienne de la fonction f , c'est-à-dire la matrice des dérivées secondes et $\frac{\partial f(\mathbf{r}_0)}{\partial \mathbf{r}}$ sa Jacobienne, c'est-à-dire le vecteur des dérivées premières.

La méthode de *Newton* consiste à commencer par une valeur initiale \mathbf{r}_0 et de calculer l'incrément de pose $\dot{\mathbf{r}}$ en utilisant l'équation (1.17).

Ce processus est répété jusqu'à la convergence, c'est-à-dire jusqu'à ce que la valeur de $\dot{\mathbf{r}}$ devienne négligeable. Cette méthode d'optimisation du second ordre rencontre moins de problèmes de minimums locaux que les méthodes du premier ordre grâce à l'utilisation de la Hessienne dans le calcul qui améliore nettement la convergence.

1.3.2 Classification des méthodes d'asservissement visuel

On peut distinguer l'asservissement visuel réel de l'asservissement visuel virtuel. Ces deux méthodes peuvent s'appliquer à l'asservissement visuel basé image (IBVS) (2D) et à l'asservissement visuel basé pose (PBVS) (3D). Le premier utilise des mesures dans l'image (primitives géométriques ou photométriques) pour le calcul de la loi de contrôle, alors que l'asservissement visuel basé pose utilise l'estimation de pose dans l'espace cartésien pour générer la loi de commande.

1.3.2.1 Asservissement visuel

a- Asservissement visuel basé primitives géométriques

L'asservissement visuel basé image vise à contrôler automatiquement un robot en se basant sur l'information perçue par un capteur, généralement une caméra montée sur le robot. Les mouvements 3D de la caméra sont générés afin de réaliser une tâche définie dans l'espace 2D : à partir d'une position initiale, définie par une image initiale, atteindre l'image désirée. Ceci revient à optimiser la similarité entre les deux images : minimiser l'erreur entre les deux images. La comparaison entre les images peut être basée sur des primitives géométriques (points [Weiss 1987] [Feddema 1989], droites [Espiau 1992] [Chaumette 1990], cercles [Espiau 1992] [Chaumette 1990],...).

Soit $\mathbf{s}(\mathbf{r})$ le vecteur des primitives de la pose courante et \mathbf{s}^* celui de la pose désirée. Le processus revient à minimiser l'erreur qui est typiquement définie par :

$$\mathbf{e} = \mathbf{s}(\mathbf{r}) - \mathbf{s}^* \quad (1.18)$$

Pour assurer la convergence de \mathbf{s} vers \mathbf{s}^* , il est nécessaire de déterminer la matrice d'interaction \mathbf{L}_s qui lie le mouvement 3D dans l'espace et le mouvement 2D des primitives dans l'image. La formulation de la matrice d'interaction en vision perspective a été initialement développée par [Feddema 1989] pour les points et par [Chaumette 1990] pour les cercles, cylindres et sphères. Cette relation est définie par [Chaumette 1990] :

$$\dot{\mathbf{s}} = \mathbf{L}_s \mathbf{v} \quad (1.19)$$

où $\dot{\mathbf{s}}$ est le mouvement des primitives dans l'image et $\mathbf{v} = (\mathbf{v}, \boldsymbol{\omega})$ est la vitesse de la caméra, avec $\mathbf{v} = (v_X, v_Y, v_Z)$ la vitesse de translation et $\boldsymbol{\omega} = (\omega_X, \omega_Y, \omega_Z)$ la vitesse de rotation. En imposant une décroissance exponentielle de l'erreur, on a :

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \quad (1.20)$$

où λ est un gain positif qui sert à définir le pas de convergence.

Ainsi, des équations (1.19) et (1.20) on obtient la loi de commande :

$$\mathbf{v} = -\lambda \mathbf{L}_s^+ \mathbf{e} \quad (1.21)$$

Tel que, \mathbf{L}_s^+ est la pseudo-inverse de la matrice d'interaction \mathbf{L}_s .

Pour un point $\mathbf{x} = (x, y)$, tel que ($\dot{\mathbf{x}} = \mathbf{L}_x \mathbf{v}$), la matrice d'interaction \mathbf{L}_x relative au point $\mathbf{x} = (x, y)$ est donnée par [Espiau 1992] :

$$\mathbf{L}_x = \begin{pmatrix} -1/Z & 0 & x/Z & xy & -(1+x^2) & y \\ 0 & -1/Z & y/Z & 1+y^2 & -xy & -x \end{pmatrix} \quad (1.22)$$

Pour obtenir des propriétés d'invariance, un autre type de primitive géométrique basé sur les moments de l'image, a été proposé dans [Chaumette 2004]. La matrice d'interaction a été calculée d'une manière analytique pour cette primitive et la loi de contrôle générée est découplée. Les moments de l'image peuvent être calculés à partir d'un ensemble de points ou de contours fermés [Tahri 2005]. Dans ces travaux, le calcul des moments ne peut se faire que sur une image binaire ou bien une image segmentée.

Dans [Hafez 2008] un ensemble de N primitives géométriques \mathbf{x} (des points 2D) a été extrait de l'image. Ces points ont été modélisés à l'aide d'un modèle de mélange

de gaussiennes. Dans ce cas, le mélange de gaussiennes est une combinaison de N fonctions de densité de probabilité qui est paramétré par un vecteur moyen $\boldsymbol{\mu}$ et une matrice de covariance $\boldsymbol{\Sigma}$. Ainsi, à partir des images courante et désirée on obtient deux mélanges de gaussiennes $f(\mathbf{x}, \mathbf{p})$ et $g(\mathbf{x})$.

Pour l'image courante :

$$f(\mathbf{x}, \mathbf{p}) = \sum_{i=1}^N \alpha_i \mathcal{N}(\mathbf{x}; \mathbf{R}\boldsymbol{\mu}_i + t, \mathbf{R}\boldsymbol{\Sigma}_i\mathbf{R}^t) \quad (1.23)$$

où \mathbf{p} est le point de vue et \mathbf{R} la matrice de rotation.

Pour l'image désirée :

$$g(\mathbf{x}) = \sum_{i=1}^N \beta_i \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i^*, \boldsymbol{\Sigma}_i^*) \quad (1.24)$$

où $\boldsymbol{\mu}_i^*$ et $\boldsymbol{\Sigma}_i^*$ sont les paramètres du mélange de gaussiennes désiré.

L'asservissement visuel est réalisé en minimisant la fonction de coût \mathcal{C} qui est la norme \mathcal{L}_2 entre les estimations des mélanges de gaussiennes à la position courante et désirée :

$$C = \int [f(\mathbf{x}, \mathbf{p}) - g(\mathbf{x})]^2 d\mathbf{p} \quad (1.25)$$

Des résultats de simulation ont été présentés pour deux d.d.l ainsi que pour les six d.d.l en utilisant l'information de profondeur. Dans le cadre des méthodes d'asservissement visuel géométrique, cette méthode se distingue par l'absence de l'étape de suivi des primitives contrairement à toutes les autres, cependant l'étape d'extraction des primitives est inévitable.

b- Asservissement visuel basé primitives photométriques

Dans les approches exposées ci-dessus, les primitives visuelles sont extraites de l'image. Ces approches ont besoin d'étapes de détection, de suivi, de mise en correspondance et enfin d'une estimation à chaque itération (chaque image) de la valeur de la primitive visuelle est nécessaire. Dans l'asservissement visuel photométrique, la primitive visuelle utilisée est l'image acquise par la caméra embarquée sur le robot. Au lieu de passer par une étape d'extraction des primitives visuelles géométriques, l'intensité des pixels de l'image est directement exploitée [Collewet 2008]. Ainsi, le vecteur \mathbf{s} est constitué des intensités des pixels de l'image. Ce type de méthode est aussi appelé asservissement visuel direct.

Le problème d'optimisation de la tâche d'asservissement visuel devient alors :

$$\hat{\mathbf{r}} = \operatorname{argmin} C(I, I^*) \quad (1.26)$$

où, C est une mesure de similarité directe entre les images courante I et désirée I^* .

Dans les premiers travaux en asservissement visuel ayant utilisé l'information photométrique [Nayar 1996] [Deguchi 2000], le principe consiste à compresser l'image en la projetant dans un sous espace constitué de valeurs propres afin de réduire la dimension des données. La loi de contrôle est appliquée dans cet espace et la matrice d'interaction est apprise en mode hors ligne. Une autre approche considérant l'intensité des pixels a été proposée dans [Kallem 2007]. Cette méthode est basée sur l'utilisation de fonctions noyaux afin de fournir une loi de contrôle découplée. Cependant, elle ne prend en compte que les trois translations et la rotation autour de l'axe optique de la caméra. Plus récemment, [Collewet 2008] a proposé une méthode qui permet d'utiliser toute l'intensité de l'image pour le calcul de la loi de contrôle. Dans ce travail la matrice d'interaction est calculée directement pendant le processus. Pour cela, la contrainte de conservation du flot optique est utilisée [Collewet 2008].

On suppose que l'intensité d'un pixel est constante suite à un mouvement :

$$I(\mathbf{x} + d\mathbf{x}, t + dt) = I(\mathbf{x}, t) \quad (1.27)$$

Sous l'hypothèse que le déplacement $d\mathbf{x}$ est très faible, le développement de Taylor au premier ordre de l'équation (1.27) au voisinage de \mathbf{x} donne :

$$\frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt = 0 \quad (1.28)$$

On note le mouvement du point dans l'image par $(\frac{dx}{dt}, \frac{dy}{dt}) = \dot{\mathbf{x}}$ et le gradient spatial de l'intensité par $\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$. Ce qui nous ramène à l'équation de conservation du flot optique :

$$\nabla I^\top \dot{\mathbf{x}} + \frac{\partial I(\mathbf{x}, t)}{\partial t} = 0 \quad (1.29)$$

où ∇I est le gradient spatial de $I(\mathbf{x}, t)$.

Ainsi la matrice d'interaction associée à l'intensité d'un pixel est donnée par :

$$\mathbf{L}_I = \nabla I^\top \mathbf{L}_x \quad (1.30)$$

où \mathbf{L}_x est la matrice d'interaction associée à un point défini par l'équation (1.22).

Cette définition de la matrice d'interaction sera utilisée tout au long de cette thèse. Notamment, dans le Chapitre 3 et le Chapitre 4 pour la définition de la loi de contrôle basée pertinence visuelle.

Les moments précédemment utilisés pour l'asservissement géométrique ont été calculés d'une autre manière en se basant sur l'information de l'intensité de l'image

[Bakthavatchalam 2013]. Cette nouvelle utilisation des moments photométriques évite les problèmes de segmentation rencontrés lors de l'utilisation des moments géométriques. Les auteurs ont calculé analytiquement la matrice d'interaction relative aux moments. Et pour le calcul de la loi de contrôle découplée, ils se sont basés sur les travaux de [Chaumette 2004] et [Tahri 2005].

Le concept d'asservissement photométrique a notamment, engendré l'asservissement visuel basé sur l'utilisation d'une fonction de similarité comme l'information mutuelle [Dame 2011], calculée à partir de l'information photométrique.

Très récemment, des travaux se basant sur les mélanges de gaussiennes ont été présentés dans le cadre de l'asservissement visuel. Plus précisément, une nouvelle modélisation de l'image à l'aide d'un mélange de gaussiennes calculé à partir de l'intensité des pixels de l'image a été proposée. L'idée de l'utilisation des mélanges de gaussiennes photométriques a vu le jour au sein de l'équipe perception robotique dont laquelle je fais partie, du laboratoire Modélisation Information et Systèmes (MIS). C'est ainsi que *Crombez et al.* [Crombez 2015] ont mis en œuvre une nouvelle primitive photométrique basée mélange de gaussiennes. Des résultats en simulation et en réel ont été présentés en contrôlant tous les d.d.l de la caméra en plus d'un paramètre agissant sur la taille des gaussiennes. Simultanément à ce travail, nous avons utilisé cette nouvelle manière d'exploiter les mélanges de gaussiennes pour proposer une nouvelle primitive : un mélange de gaussiennes basées saillance pour faire du cadrage visuel. Cette contribution sera détaillée dans le Chapitre 3.

Dans le cadre de nos travaux nous voulons exploiter le mécanisme d'asservissement visuel dans le but de réaliser une tâche de positionnement dans un environnement mais sans avoir une image désirée. En animation virtuelle, [Courty 2001] ont proposé d'effectuer des tâches de positionnement de la caméra répondant à certains critères : comme la maximisation de luminance, l'évitement d'obstacles et d'occultations [Berger 1997].

c- Asservissement visuel basé pose (3D)

En asservissement visuel basé pose [Wilson 1996] [Martinet 1999], la primitive agissant dans la boucle d'asservissement n'est pas obtenue à partir du plan image mais de l'espace cartésien. L'information exploitée n'est autre que la pose 3D de l'objet par rapport à la caméra. Ainsi, d'après [Chaumette 2006], si on note par \mathcal{R}_c le repère de la caméra à la pose courante, \mathcal{R}_{c^*} le repère de la caméra à la pose désirée et par \mathcal{R}_o le repère objet, le vecteur \mathbf{s} courant est constitué des vecteurs de translations et de rotations entre la pose courante et la pose désirée :

$$\mathbf{s} = ({}^{c^*}\mathbf{t}_c, \theta \mathbf{u}) \quad (1.31)$$

et le vecteur de primitive désiré $\mathbf{s}^* = (0, 0)$. Ainsi, l'erreur qu'on cherche à annuler est $\mathbf{e} = \mathbf{s}$. La matrice d'interaction est définie par :

$$\mathbf{L}_s = \begin{pmatrix} {}^{c^*}\mathbf{R}_c & 0 \\ 0 & \mathbf{L}_{\theta\mathbf{u}} \end{pmatrix} \quad (1.32)$$

tel que, $\mathbf{L}_{\theta\mathbf{u}}$ est donnée par [Malis 1999] :

$$\mathbf{L}_{\theta\mathbf{u}} = \mathbf{I}_3 - \frac{\theta}{2}[\mathbf{u}]_{\times} + \left(1 - \frac{\text{sinc } \theta}{\text{sinc}^2 \frac{\theta}{2}}\right) [\mathbf{u}]_{\times}^2 \quad (1.33)$$

où *sinc* est le sinus cardinal défini par $\text{sinc } x = x$ et $\text{sinc } 0 = 1$.

La loi de contrôle obtenue est découplée dans la mesure où :

$$\begin{cases} v = -\lambda {}^{c^*}\mathbf{R}_c^{\top} {}^{c^*}\mathbf{t}_c \\ \omega = -\lambda \theta \mathbf{u} \end{cases} \quad (1.34)$$

Cette loi de contrôle permet d'estimer parfaitement les paramètres de la caméra pour se déplacer d'une pose à une autre sans tenir compte du contenu visuel des images engendrées par le déplacement.

Dans cette thèse, la tâche de déplacement de la caméra virtuelle, d'une pose initiale à une pose désirée, est traitée par asservissement visuel basé pose, en utilisant les d.d.l de translations uniquement. Alors que les d.d.l de rotations sont contrôlés à l'aide d'une primitive basée image afin de garantir la pertinence de contenu de l'image. Ce mécanisme est expliqué dans le Chapitre 4.

1.3.2.2 Asservissement visuel virtuel

Les mêmes techniques développées pour la robotique peuvent être étendues à la navigation dans un environnement virtuel [Courty 2001] [Marchand 2002a]. La seule différence entre l'asservissement visuel (AV) et l'asservissement visuel virtuel (AVV) est qu'on passe à la fois d'un monde réel à un monde virtuel (maillage, nuage de points 3D dense) et d'une caméra réelle embarquée sur un robot à une caméra virtuelle. Le processus de résolution adopté est similaire, dans la mesure où les primitives visuelles de l'image courante et de l'image désirée sont utilisées pour minimiser l'erreur. Dans cette thèse, nous allons exploiter les principes de l'AVV pour réaliser la tâche de navigation dans un environnement virtuel. Cependant, dans notre démarche aucune image désirée n'est définie. La fonction de coût modélise un critère à optimiser pour déplacer la caméra.

1.3.2.3 Prise en compte de tâches supplémentaires

L'objectif ici est que la loi de commande calculée par l'asservissement visuel prenne en compte la réalisation de plusieurs tâches en parallèle. Dans la littérature, ce concept de combinaison de plusieurs tâches a été résolu de deux manières différentes.

a- Combinaison de primitives

Parmi les méthodes de combinaison de tâches on retrouve la méthode hybride qui consiste à empiler les matrices d'interaction et les vecteurs d'erreur associés aux différentes tâches à réaliser.

Cette méthode [Courty 2002] hybride a été utilisée pour la tâche de cadrage d'un tableau. Pour ce faire, le contrôle de la caméra a été réalisé sur tous ses d.d.l en cherchant à positionner un rectangle précisément dans l'image. Ce rectangle est donc représenté par les coordonnées de ses quatre cotés $\mathbf{P}_i = (x_i, y_i)$ ($i = 1..4$) et sa matrice d'interaction par :

$$\mathbf{L} = [\mathbf{L}_{\mathbf{P}_1}, \mathbf{L}_{\mathbf{P}_2}, \mathbf{L}_{\mathbf{P}_3}, \mathbf{L}_{\mathbf{P}_4}]^T \quad (1.35)$$

où $\mathbf{L}_{\mathbf{P}_i}$ ($i = 1..4$) sont les matrices d'interaction relatives à chaque coté du rectangle, définies par l'équation (1.22).

Le vecteur d'erreur associé au rectangle est donné par :

$$(\mathbf{s} - \mathbf{s}^*) = (\mathbf{P}_i - \mathbf{P}_{di}) = \begin{pmatrix} x_1 - x_{d1} \\ y_1 - y_{d1} \\ x_2 - x_{d2} \\ y_2 - y_{d2} \\ x_3 - x_{d3} \\ y_3 - y_{d3} \\ x_4 - x_{d4} \\ y_4 - y_{d4} \end{pmatrix} \quad (1.36)$$

où \mathbf{P}_{di} sont les positions désirées des quatre points dans l'image.

Ainsi, on obtient la loi de contrôle finale permettant le cadrage du tableau :

$$\mathbf{v} = -\mathbf{L}^+(\mathbf{s} - \mathbf{s}^*) \quad (1.37)$$

La figure 1.4 montre quelques images clés rendues durant la tâche du cadrage du tableau.

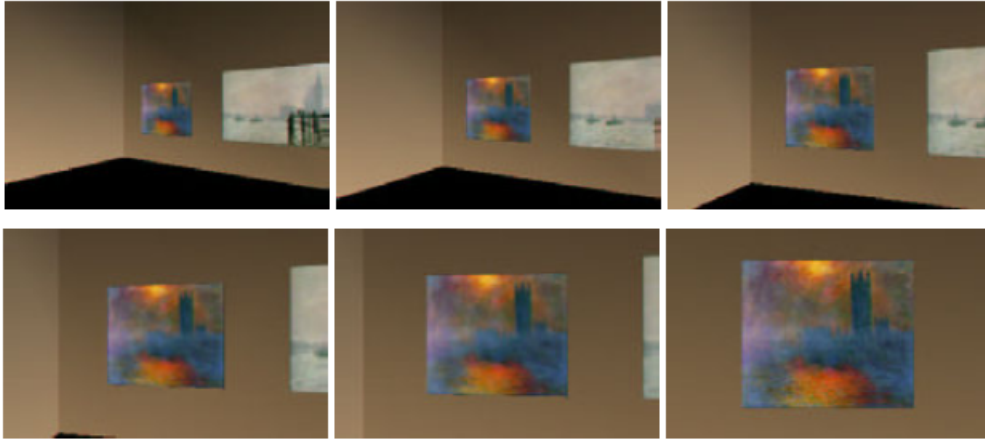


FIGURE 1.4 – Séquence d’images relative au cadrage d’un tableau [Courty 2002].

b- Le formalisme de redondance

Le formalisme de redondance [Rosen 1961, Chiaverini 1997] permet de réaliser une tâche principale prioritaire et une ou plusieurs autres tâches secondaires tout en évitant les conflits.

Considérons \mathbf{e}_1 et \mathbf{e}_2 deux fonctions de tâches, avec \mathbf{J}_1 et \mathbf{J}_2 leurs Jacobiens respectifs et \mathbf{r} la pose de la caméra. Le Jacobien de la tâche \mathbf{e}_i ($i = 1, 2$) $\mathbf{J}_i = \frac{\partial \mathbf{e}_i}{\partial \mathbf{r}}$ est défini par l’équation suivante :

$$\dot{\mathbf{e}}_i = \frac{\partial \mathbf{e}_i}{\partial \mathbf{r}} \dot{\mathbf{r}} \quad (1.38)$$

Les Jacobiens des tâches sont de rang plein. Ainsi, si la taille du Jacobien est $n \times m$ alors son rang est égal à m . La caméra est contrôlée par la loi de commande \mathbf{v} (qui correspond à l’incrément de pose $\dot{\mathbf{r}}$), qui est obtenu en inversant l’équation 1.38 :

$$\mathbf{v} = \mathbf{J}_i^+ \dot{\mathbf{e}}_i^* \quad (1.39)$$

où $\dot{\mathbf{e}}_i^*$ est le mouvement souhaité de la caméra dans l’espace de tâche.

A titre d’exemple, pour les deux tâches \mathbf{e}_1 et \mathbf{e}_2 nous pouvons utiliser la loi suivante :

$$\mathbf{v} = \mathbf{J}_1^+ \dot{\mathbf{e}}_1^* + (\mathbf{I} - \mathbf{J}_1^+ \mathbf{J}_1) \dot{\mathbf{e}}_2^* \quad (1.40)$$

où \mathbf{J}_1^+ est la pseudo-inverse de la matrice \mathbf{J}_1 et $(\mathbf{I} - \mathbf{J}_1^+ \mathbf{J}_1)$ est le projecteur vectoriel sur le noyau du Jacobien \mathbf{J}_1 . Ce projecteur assure que l’exécution de la tâche \mathbf{e}_2 n’affecte pas celle de la tâche \mathbf{e}_1 . Ceci est équivalent à exécuter la tâche principale \mathbf{e}_1 et la partie de la tâche \mathbf{e}_2 qui ne se trouve pas dans le noyau de la tâche \mathbf{e}_1 .

Ce formalisme a été proposé par *Nelson et al.* pour minimiser une fonction objective. Cette fonction combine la tâche visuelle permettant le suivi d'une cible à l'aide d'une caméra montée sur un robot avec d'autres contraintes, comme l'évitement de singularités cinématiques, l'évitement des limites des articulations ou une contrainte sur le champ de vue [Nelson 1994]. Une approche similaire a été utilisée dans [Marchand 2002b] pour planifier la trajectoire de la caméra tout en évitant des configurations indésirables comme la collision avec des obstacles.

Dans [Marchand 2007], l'auteur a proposé une méthode en exploitant le formalisme de redondance pour contrôler le positionnement de la caméra ou de la lumière afin d'observer l'objet à partir du meilleur point de vue. Ce problème peut être résolu de deux manières différentes, soit en bougeant la caméra, soit en bougeant la lumière. Nous allons nous limiter à la présentation de la première partie.

Le but de la démarche est de positionner la caméra de façon à cadrer l'objet considéré en maximisant la luminance.

Pour le cadrage de l'objet, une tâche d'asservissement visuel classique est considérée. La primitive visuelle à la pose courante est le centre de gravité de l'objet $\mathbf{s} = (u, v)$, alors que la primitive à la pose désirée est $\mathbf{s}^* = (0, 0)$.

Pour la contrainte de maximisation de luminance, une fonction de coût constituée de la somme des intensités des pixels de l'image est utilisée :

$$h_s = \sum_u \sum_v I(u, v) \quad (1.41)$$

où $I(u, v)$ est l'intensité du pixel de coordonnées (u, v) .

La variation de cette fonction de coût due au mouvement de la caméra est donnée par son gradient :

$$\frac{\partial h_s(\mathbf{r})}{\partial \mathbf{r}} = \sum_u \sum_v \nabla I^\top \mathbf{L}_u \quad (1.42)$$

où \mathbf{r} est la pose de la caméra, ∇I est le gradient spatial de l'image et \mathbf{L}_u est la matrice d'interaction liée au pixel $\mathbf{u} = (u, v)$.

Ainsi, en utilisant le formalisme de redondance, la fonction de tâche qui prend en compte le positionnement de la caméra tout en maximisant la luminance est donnée par :

$$\mathbf{e} = \mathbf{L}_s^+(\mathbf{s} - \mathbf{s}^*) + (\mathbf{I} - \mathbf{L}_s^+ \mathbf{L}_s) \frac{\partial h_s(\mathbf{r})}{\partial \mathbf{r}} \quad (1.43)$$

Une validation de l'approche a été effectuée positionnant la caméra par rapport à la sphère comme on peut le voir sur la figure 1.5. On remarque que la luminance

de la sphère augmente en partant de l'image de gauche vers l'image, de droite, tout en gardant l'objet centré dans l'image.

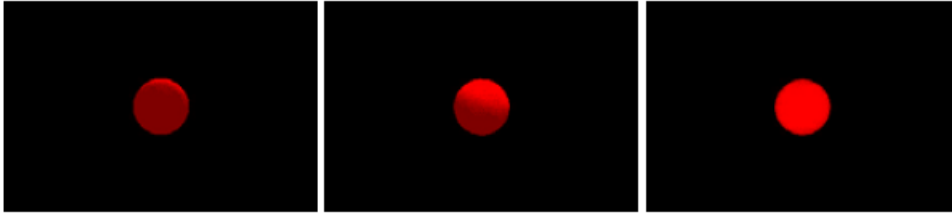


FIGURE 1.5 – Positionnement de la caméra par rapport à une shpère en maximisant la luminance [Marchand 2007].

1.4 Conclusion

Dans ce chapitre, quelques principes géométriques et mathématiques relatifs à la vision par ordinateur ont été introduits. Nous avons aussi présenté les différents types d'asservissement visuel : l'asservissement visuel basé sur des primitives géométriques, qui nécessite une étape de mise en correspondance ; l'asservissement visuel direct qui utilise l'intensité des pixels de l'image ; et l'asservissement visuel basé pose qui utilise l'information 3D. Notre objectif est d'exploiter le schéma d'asservissement visuel virtuel direct dans le but de réaliser une tâche de navigation répondant à quelques critères que nous allons introduire dans le chapitre 2.

Assistance à la navigation : outils et méthodes

Sommaire

2.1	Introduction	25
2.2	Pertinence d'un point de vue	26
2.2.1	Méthodes photométriques	27
2.2.2	Méthodes géométriques	28
2.3	Contrôle de caméra virtuelle	34
2.3.1	Méthodes manuelles et assistées	35
2.3.2	Méthodes automatiques	38
2.4	Conclusion	42

2.1 Introduction

Avec l'apparition des modèles 3D de plus en plus complexes et la sophistication des jeux vidéos, la navigation virtuelle rencontre beaucoup de difficultés. La cause peut être due à la quantité de données à traiter ou à la structuration complexe des environnements virtuels, et ceci malgré l'utilisation de puissants composants graphiques. C'est la raison pour laquelle les outils d'assistances à la navigation ont vu le jour, pour permettre à l'utilisateur d'interagir intuitivement et naturellement dans ce type d'environnements.

L'assistance à la navigation regroupe deux aspects qui sont la navigation et l'assistance à l'interaction. La navigation consiste à agir sur les paramètres de la caméra virtuelle (position, orientation et facteurs d'échelle). Cette modification peut se faire automatiquement en utilisant des algorithmes appropriés ou, elle peut être due à un contrôle réalisé par l'utilisateur à l'aide de dispositifs d'interaction (clavier, souris, joystick,...). Cependant, les périphériques dédiés à l'interaction ne peuvent pas être utilisés directement, mais accompagnés d'un système d'assistance pour

rendre aisée la tâche de navigation. En effet, l'utilisateur ne devrait pas avoir à fournir d'effort pour s'adapter au système de navigation virtuelle. C'est le système qui devrait être en mesure de répondre à ses attentes pour lui garantir une immersion complète dans le monde virtuel.

En vision par ordinateur l'assistance à la navigation est considérée comme un problème de contrôle de caméra virtuelle. Ce domaine est très vaste et englobe différents domaines et applications. Dans ce chapitre nous allons commencer par aborder les méthodes existantes qui permettent d'évaluer la qualité d'un point de vue de la caméra pour visualiser un objet ou une scène virtuelle. Les méthodes basées sur la théorie de l'information et sur l'attention visuelle vont être traitées plus en détail, étant donné leur utilisation dans cette thèse pour la définition de la pertinence visuelle. Dans la littérature, ces méthodes se basent généralement sur des primitives géométriques relatives au type de modèle 3D considéré (maillage, nuage de points). Dans nos travaux, nous nous appuyons sur des primitives photométriques de l'image rendue à partir d'un environnement virtuel ou à partir d'une caméra réelle montée sur un robot. Ensuite, nous allons présenter une classification regroupant quelques méthodes de contrôle de caméra.

2.2 Pertinence d'un point de vue

Bien que la mesure de la qualité ou la pertinence d'un point de vue soit une question plus au moins subjective, plusieurs travaux scientifiques [Plemenos 1996, Barral 2000, Vázquez 2001] s'y sont intéressés pour tenter de la modéliser, en posant l'hypothèse qu'un point de vue pertinent est celui à partir duquel une scène virtuelle peut être comprise. En effet, la pertinence d'un point de vue dépend de la quantité d'information perçue à partir de celui-ci. Elle peut être de type photométrique ou géométrique.

Dans le cadre des approches photométriques, la pertinence a été mesurée en utilisant la luminance [Courty 2001] et la saillance [Itti 1998].

Parmi les méthodes géométriques on retrouve :

- Les méthodes qui calculent le nombre de faces visibles [Plemenos 1996], [Stuerzlinger 1999].
- Les méthodes qui évaluent la surface totale des faces projetées sur une sphère unitaire centrée sur le point de vue [Barral 2000].
- Les méthodes qui estiment l'angle de vue par rapport à l'orientation des faces [Kamada 1988].
- Les méthodes qui exploitent la théorie de l'information et plus précisément

l'entropie de Shannon [Vázquez 2001].

- Les méthodes qui calculent la saillance 3D en utilisant des descripteurs géométriques [Shtrom 2013].

2.2.1 Méthodes photométriques

En vue de mesurer la qualité d'un point de vue, les méthodes photométriques exploitent directement l'information d'intensité de l'image [Courty 2001] ainsi que les algorithmes de l'attention visuelle pour le calcul de la saillance [Itti 1998].

Les systèmes d'attention visuelle sont utilisés pour calculer les parties intéressantes d'une image ou d'une vidéo donnée. Ce genre d'algorithmes se révèle être pratique pour réduire la quantité d'information qu'un système de vision par ordinateur a besoin d'analyser. Ainsi, dans le but de réduire la complexité d'une tâche, seules les informations les plus importantes sont prises en compte. Dans les approches "ascendantes" des systèmes d'attention visuelle, on trouve les algorithmes de saillance qui ont longtemps été utilisés pour la détection d'objets dans des environnements encombrés.

Le système d'attention visuelle le plus connu est basé sur des modèles biologiques [Itti 1998]. Ces modèles s'inspirent du système visuel humain qui n'exploite qu'une partie de ce qui est visible pour l'analyse d'une scène complexe. C'est ce qui est connu sous le nom de "foyer d'attention" [Treisman 1980]. La rétine de l'œil humain contient des cellules ganglionnaires qui reçoivent les informations visuelles à partir des récepteurs d'images à travers les cellules bipolaires. Le champ de réception des cellules ganglionnaires est composé de deux zones, l'entourage et le centre. Il existe deux types (Figure 2.1) de cellules ganglionnaires [Palmer 1999] :

- La cellule ganglionnaire "On-centre" qui correspond à des zones lumineuses entourées par un fond sombre.
- La cellule ganglionnaire "Off-center" qui correspond à des zones sombres entourées par un fond lumineux.

Les systèmes d'attention visuelle se sont inspirés de la composition de ces cellules pour le calcul de la saillance en remplaçant les cercles des cellules par des pixels. Notamment les systèmes "ascendants" de l'attention visuelle [Itti 1998, Bruce 2005, Harel 2006]. Ces approches sont réalisées en trois étapes. En premier lieu arrive l'étape d'extraction des primitives de bas niveau comme l'intensité, la couleur, l'orientation et la texture de l'image initiale qui est sous-échantillonnée à différentes échelles sous forme de cartes. Ensuite, le calcul de la saillance est effectué soit en calculant le "centre-entourage" [Itti 1998] sur chaque carte, en maximisant

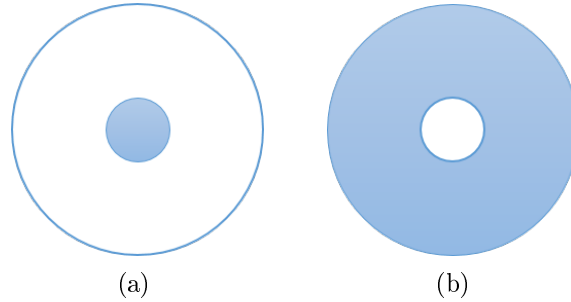


FIGURE 2.1 – (a) off-center des cellules ganglionnaires, (b) on-center des cellules ganglionnaires.

l'information d'une scène à l'aide de l'entropie de Shannon [Bruce 2005], ou basée sur le calcul de graphes [Harel 2006]. Et finalement, une combinaison linéaire ou non-linéaire accompagnée d'une normalisation ou non de ces cartes est réalisée pour obtenir la carte de saillance finale.

Dans cette thèse (Chapitre 3), le calcul de l'image de saillance que nous avons utilisé exploite une approche [Radhakrishna 2009] qui estime le contraste en combinant des primitives de couleur et de luminance.

Le calcul de la carte de saillance S pour une image I est exprimé par :

$$S(u, v) = \|I_\mu - I_f(u, v)\| \quad (2.1)$$

où I_f est l'image filtrée par un flou Gaussien (en utilisant une fenêtre de taille 5×5) et I_μ est la moyenne des intensités des pixels pour la fenêtre centrée sur le pixel (u, v) . $\|\cdot\|$ représente la norme \mathcal{L}_2 calculée à partir de la distance euclidienne. Un exemple de résultat est illustré dans la Figure 2.2.

D'autres méthodes peuvent être utilisées. Cependant, nous avons choisi celle-ci pour la qualité et la précision des résultats dans la mesure où les zones saillantes sont bien définies et segmentées dans la carte de saillance. Elle permet notamment de mettre l'accent sur les plus grands objets saillants et met en évidence d'une manière uniforme toutes les régions saillantes et non simplement leurs contours. En plus de cela, le temps de calcul est faible (environ 0.1 seconde pour une image de taille 255×145).

2.2.2 Méthodes géométriques

Dans le même contexte de mesure de la qualité d'un point de vue, on retrouve les approches qui calculent le nombre de faces perçues à partir d'un certain point

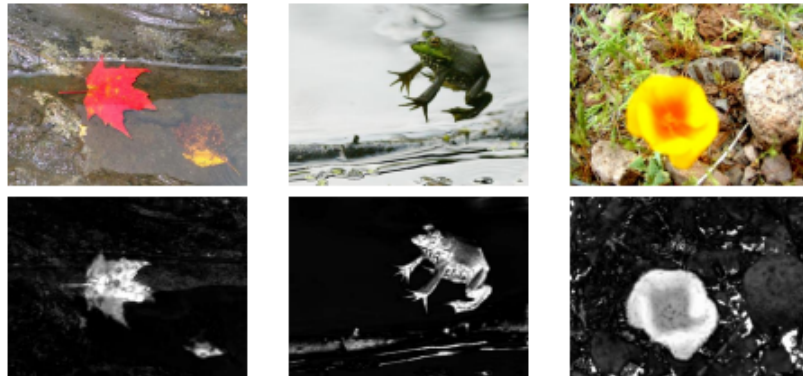


FIGURE 2.2 – Cartes de saillance calculées en utilisant [Radhakrishna 2009].

de vue. [Plemenos 1996] ont utilisé cette méthode pour l'élaboration d'une mesure de qualité, car elle constitue un critère qui peut informer sur le nombre de détails présents dans une scène. Cependant, ce critère semble être insuffisant, dans la mesure où il est relatif à la taille physique des faces. En effet, si on a dans un même modèle une discontinuité par rapport à la taille des faces, plusieurs faces de petite taille engendrent moins d'information qu'un nombre plus restreint de faces mais de taille nettement plus grande.

[Barral 2000] ont proposé d'utiliser la somme des surfaces projetées comme mesure de qualité en s'appuyant sur le critère suivant. Plus la surface projetée est large plus on perçoit d'informations. Néanmoins, à partir de deux points de vue différents engendrant deux surfaces projetées de tailles égales, la quantité d'information observée peut être différente. En effet, la surface projetée de l'un des points de vue peut provenir de la même face alors qu'à partir de l'autre point de vue elle peut être relative à plusieurs faces.

Une autre approche de mesure de qualité s'appuie sur le calcul de l'angle entre la direction du point de vue et la normale de la face. [Kamada 1988] ont constaté que dans le cas d'une projection orthographique, cet angle aide à déterminer la manière dont une face est rendue. Cette approche a des limitations et engendre des mesures erronées dans le cas où il y a des occultations. En plus, elle ne tient pas compte de la distance entre le point de vue et la face (surface de projection très petite, quantité d'information perçue négligeable malgré la valeur adéquate de l'angle entre la direction du point de vue et la normale de la face). [Dorme 2001] ont ajouté d'autres paramètres pour résoudre ces limitations, qui consistent à pondérer l'angle de vue avec le nombre de faces perçues et la taille de la surface projetée. Cependant, cette approche peut s'avérer défaillante car la pondération qui doit être

bien estimée pour chaque objet peut donner la même importance à des faces de petite taille et d'autres de plus grande taille.

[Vázquez 2001] se sont intéressés à une approche mathématique pour le calcul de la qualité d'un point de vue. Ils ont utilisé la théorie de l'information et plus précisément l'entropie de Shannon. Cette approche permet de quantifier le taux d'information géométrique perçue à partir d'un point de vue en utilisant aussi le critère des surfaces projetées sur une sphère centrée sur le point de vue. Ils ont pris en compte aussi la surface du fond dans le cas des scènes ouvertes dans la mesure où ce dernier représente une source d'information.

Pour une surface S et un point de vue \mathbf{p} , l'entropie d'un point de vue (Figure 2.3) est définie par :

$$E(S, \mathbf{p}) = - \sum_{i=0}^{N_f} \frac{A_i}{A_t} \log_2 \frac{A_i}{A_t} \quad (2.2)$$

où N_f est le nombre de faces de la scène, A_i la surface de la face i projetée sur la sphère, A_0 représente la surface du fond projetée sur la sphère dans le cas des scènes ouvertes ($A_0 = 0$ dans le cas des scènes fermées) et A_t la surface totale de la sphère. Ainsi le rapport $\frac{A_i}{A_t}$ représente la visibilité de la face i par rapport au point de vue \mathbf{p} .

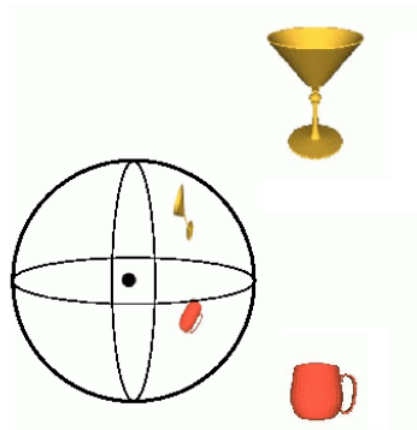


FIGURE 2.3 – L'entropie d'un point de vue (illustration de la surface projetée sur la sphère) [Vázquez 2001].

En utilisant presque la même formulation de l'entropie introduite par [Vázquez 2001], sans prendre en compte le fond, [Gumhold 2002] a pu estimer la quantité d'information de luminance présente dans une scène 3D. Ces approches ne peuvent être utilisées que pour des scènes 3D maillées et donc ne sont pas applicables à un nuage de points 3D dense.

D'autre part, les algorithmes de calcul de saillance visuelle ont été étendus à des environnements 3D, où le but est de calculer la saillance d'un maillage ou d'un nuage de points 3D. [Lee 2005] ont introduit une mesure d'importance de régions pour des maillages afin de déterminer les régions visuellement intéressantes dans un objet maillé (Figure 2.4). Pour le calcul de la saillance de l'objet, un opérateur "centre-entourage" sur des primitives géométriques (les courbures moyennes pondérées avec une gaussienne) a été utilisé.

Un autre type de saillance 3D, cette fois considérant des nuages de points extrêmement larges (Figure 2.5) a été proposé par [Shtrom 2013]. L'algorithme est basé sur un calcul de dissimilarité entre les descripteurs points FPFH (Fast Point Feature Histogram), et le calcul de saillance est réalisé d'une manière hiérarchique en combinant trois mesures de calcul de cette variance sur différents voisinages.

Le but est de trouver les sommets qui ont un voisinage unique géométriquement parlant. En effet, un sommet est considéré distinct si son descripteur est différent de tous les descripteurs de l'entourage considéré. Pour un sommet \mathbf{x}_i , d'index i , le calcul de sa saillance 3D est exprimé de la manière suivante :

$$S(\mathbf{x}_i) = \frac{1}{2} (\mathcal{D}_{low}(\mathbf{x}_i) + \mathcal{A}_{low}(\mathbf{x}_i)) + \frac{1}{2} \mathcal{D}_{high}(\mathbf{x}_i) \quad (2.3)$$

Le terme \mathcal{D}_{low} permet de détecter les caractéristiques fines et il est exprimé par :

$$\mathcal{D}_{low}(\mathbf{x}_i) = 1 - \exp \left(-\frac{1}{N_x} \sum_{\mathbf{x}_j \in N_x} d_L(\mathbf{x}_i, \mathbf{x}_j) \right) \quad (2.4)$$

tel que, d_L est la mesure de dissimilarité calculée à partir de la distance euclidienne

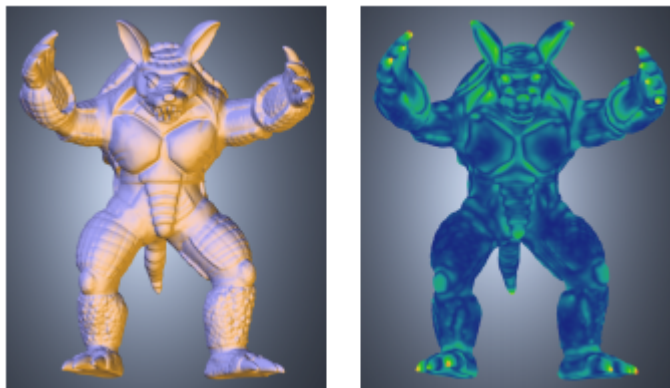


FIGURE 2.4 – Résultat du calcul de la saillance sur un maillage (les zones vertes et jaunes sont les plus saillantes) [Lee 2005].

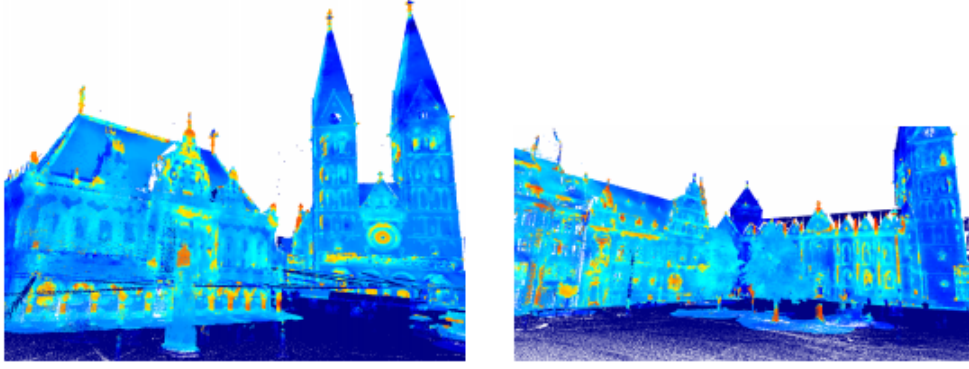


FIGURE 2.5 – Résultat du calcul de la saillance sur un nuage de points dense (les zones jaunes, oranges et rouges sont les plus saillantes) [Shtrom 2013].

et de la dissimilarité basée sur la loi χ^2 :

$$d_L(\mathbf{x}_i, \mathbf{x}_j) = \frac{D_{\chi^2}(\mathbf{x}_i, \mathbf{x}_j)}{1 + \|\mathbf{x}_i - \mathbf{x}_j\|} \quad (2.5)$$

et N_x est un voisinage où un ensemble de sommets ont une valeur de D_{χ^2} inférieure à un certain seuil. Le seuil choisi est faible ce qui engendre un voisinage de petite taille.

La mesure de dissimilarité basée sur la loi χ^2 est calculée à partir des descripteurs géométriques FPFH de chaque sommet, modélisée sous la forme d'un histogramme :

$$D_{\chi^2}(\mathbf{x}_i, \mathbf{x}_j) = \sum_{n=0}^N \frac{(FPFH_n(\mathbf{x}_i) - FPFH_n(\mathbf{x}_j))^2}{FPFH_n(\mathbf{x}_i) + FPFH_n(\mathbf{x}_j)} \quad (2.6)$$

où N est le nombre de classes de l'histogramme et $FPFH_n$ est la valeur de la nième classe de l'histogramme. Plus de détails concernant le calcul du descripteur FPFH sont présents dans [Shtrom 2013].

Le deuxième terme \mathcal{A}_{low} relatif au calcul de la saillance 3D permet de supprimer les sommets isolés résultant du calcul de \mathcal{D}_{low} .

Environ 20% des sommets qui ont une valeur élevée de \mathcal{D}_{low} vont être considérés comme des sommets principaux. Ainsi, pour le calcul de ce deuxième terme, ces sommets principaux seront considérés. Soit \mathbf{x}_{fi} le sommet principal le plus proche (en terme de distance) du sommet \mathbf{x}_i et $\mathcal{D}_{foci}(\mathbf{x}_i)$ la valeur de \mathcal{D}_{low} obtenue pour le sommet \mathbf{x}_{fi} , alors la valeur de \mathcal{A}_{low} pour le sommet \mathbf{x}_i est :

$$\mathcal{A}_{low}(\mathbf{x}_i) = \mathcal{D}_{foci}(\mathbf{x}_i) \exp\left(-\frac{\|\mathbf{x}_{fi} - \mathbf{x}_i\|^2}{2\sigma^2}\right) \quad (2.7)$$

où σ est un scalaire fixé empiriquement à 0,05 dans [Shtrom 2013].

Finalement, le troisième terme \mathcal{D}_{high} pour le calcul de cette saillance 3D permet de détecter les zones de saillance plus étendues en considérant un voisinage plus large N_x :

$$\mathcal{D}_{high}(\mathbf{x}_i) = 1 - \exp\left(-\frac{1}{N_x} \sum_{\mathbf{x}_j \in N_x} d_H(\mathbf{x}_i, \mathbf{x}_j)\right) \quad (2.8)$$

tel que,

$$d_H(\mathbf{x}_i, \mathbf{x}_j) = D_{\chi^2}(\mathbf{x}_i, \mathbf{x}_j) \log(1 + \|\mathbf{x}_i - \mathbf{x}_j\|) \quad (2.9)$$

Le logarithme est utilisé afin de diminuer l'influence des sommets qui sont dans un voisinage très proche, sans pour autant affecter ceux qui se trouvent dans un voisinage plus large.

Cette méthode de calcul de saillance 3D est valide pour tout nuage de points 3D. C'est pour cela que nous l'avons implémentée et nous l'avons testée sur les mêmes données que [Shtrom 2013].

Dans les Figures 2.6 et 2.7, on observe en moyenne les mêmes zones de saillances (jaune, orange et rouge) dans les nuages considérés. La différence des résultats est certainement due au mode d'affichage et à la taille des voisinages considérée pour le calcul des termes nécessaires pour la saillance 3D, qui n'est pas spécifiée pour chaque nuage dans cette publication [Shtrom 2013].

Nous avons appliqué cette méthode dans le but de visualiser les zones de saillance dans le modèle 3D de la cathédrale d'Amiens (Figure 2.8), en vue de réaliser une navigation automatique basée saillance 3D. Cependant, le résultat obtenu sur le nuage

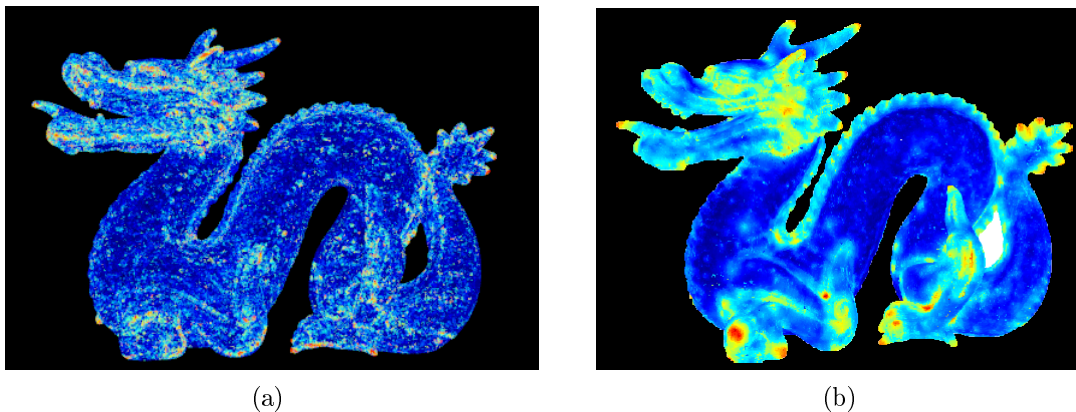


FIGURE 2.6 – (a) Résultat obtenu avec notre implémentation, (b) Résultat obtenu dans [Shtrom 2013].

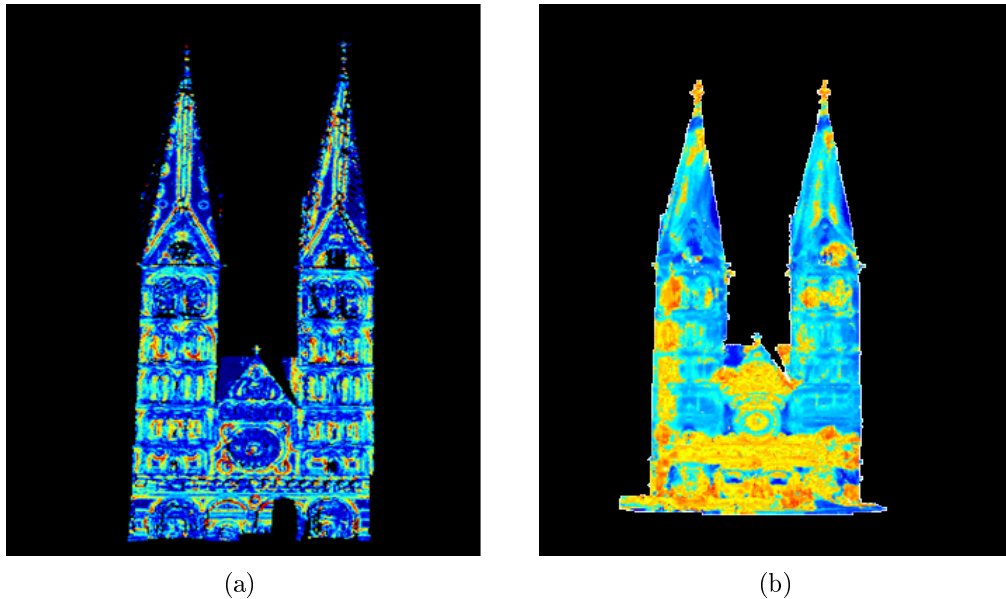


FIGURE 2.7 – (a) Résultat obtenu avec notre implémentation, (b) Résultat obtenu dans [Shtrom 2013].

de l'intérieur de la cathédrale est difficile à exploiter. Cette méthode n'utilise que la géométrie du modèle pour le calcul. Étant donné que le modèle de l'intérieur de la cathédrale est très complexe, les zones de saillance sont éparpillées sur une grande partie du modèle, notamment, des zones non pertinentes visuellement comme les chaises. Une perspective concernant ce travail consisterait à modifier le descripteur utilisé pour le calcul de la saillance en incluant l'information photométrique (couleur des sommets par exemple).

2.3 Contrôle de caméra virtuelle

L'exploration des environnements virtuels 3D est devenue un sujet récurrent dans la littérature. C'est donc tout naturellement qu'on retrouve un large éventail de techniques dédiées au contrôle des caméras dans ces environnements. Nous distinguons les méthodes destinées à l'exploration ou la navigation de celles qui concernent la manipulation d'objets 3D. Dans cette section, une plus forte attention est consacrée aux méthodes qui visent la navigation, qui sont classifiées en méthodes manuelles, assistées et automatiques.

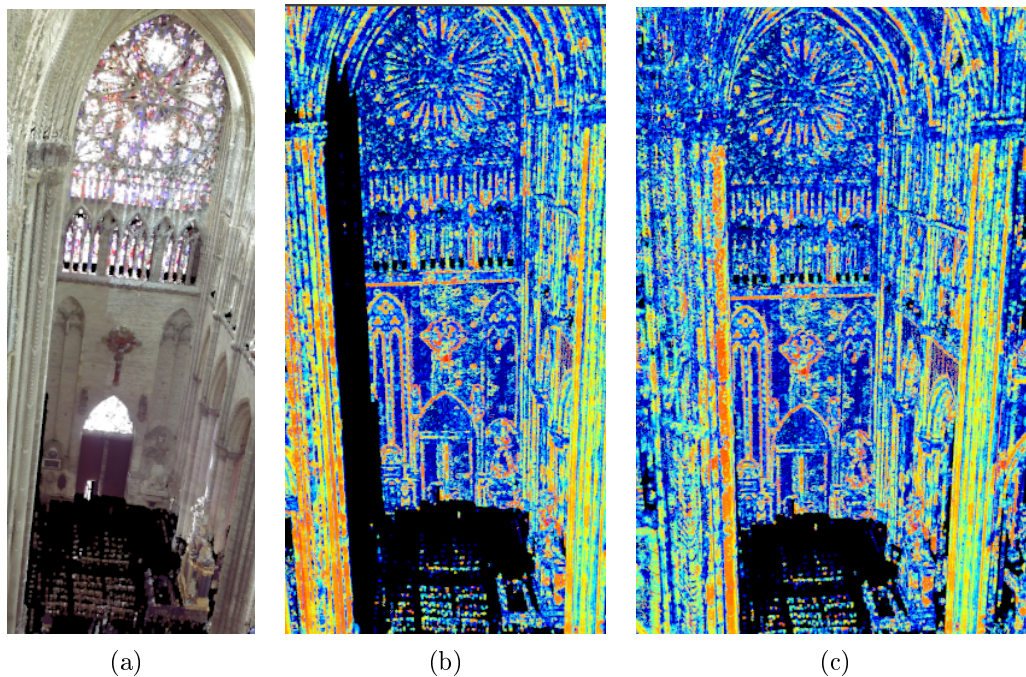


FIGURE 2.8 – (a)Extrait du nuage intérieur de la cathédrale, (b) Saillance 3D : point de vue 1, (c) Saillance 3D : point de vue 2.

2.3.1 Méthodes manuelles et assistées

Le contrôle manuel ou direct de la caméra virtuelle consiste à agir directement sur les d.d.l de la caméra afin de la déplacer (translations, panorama, zoom) (Figure 2.9) dans une scène virtuelle. Pour cela, on utilise un dispositif d'interaction afin de changer le point de vue de la caméra. Or, les six d.d.l de la caméra (trois translations suivant les axes \vec{X} , \vec{Y} et \vec{Z} , et trois rotations autour de ces axes) rendent cette tâche difficile. Par ailleurs, dans le cas de la navigation à grande échelle, l'utilisateur peut facilement se désorienter. Dans le but de réduire cette difficulté *Ware et al.* [Osborne 1990] ont proposé des métaphores qui couvrent un large panel de techniques d'interaction.

La métaphore "monde en main" : La position et l'orientation de la caméra sont fixées. En revanche, c'est le "monde" (l'objet ou la scène) qui va être déplacé en concordance avec l'action de l'utilisateur. Ce type d'interaction est très utilisé pour l'inspection des objets en détail. Cette métaphore est implémentée dans plusieurs logiciels récents comme *Blender*, *Maya* de *Autodesk*, *Cyclone* de *Leica geosystems* où les techniques de rotation s'inspirent des travaux de [Chen 1988] et aussi de

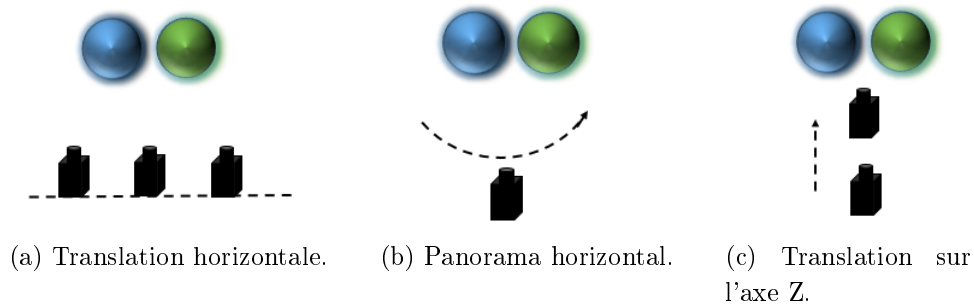


FIGURE 2.9 – Quelques types de mouvements de caméra.

[Shoemake 1992]. Cette technique considère qu'il y a une sphère virtuelle contenant l'objet à inspecter. Ainsi, l'axe et l'angle de rotation sont calculés en projetant les coordonnées de la souris sur la sphère.

La métaphore "œil en main" : La caméra est directement manipulée par l'utilisateur. Ce type de contrôle est très fréquent dans les jeux vidéo comme *Wolfenstein enemy territory* (Figure 2.10), *Mad Max* et *GTA*, notamment pour se déplacer soit en marchant ou en conduisant un véhicule.



FIGURE 2.10 – Copie écran du jeu Wolfenstein enemy territory (métaphore "œil en main").

La métaphore "vol libre" : La caméra est considérée comme un appareil volant. Ce type d'approche est très efficace pour l'exploration de vastes environnements

virtuels. Cependant, cette technique reste difficile à exploiter car l'utilisateur peut facilement se perdre durant l'exploration surtout s'il s'agit d'une scène complexe. Ce problème a été abordé par [Turner 1991] en restreignant la liberté de l'utilisateur et cela en réduisant les d.d.l de la caméra. [Tan 2001] ont proposé une technique qui couple la vitesse de vol et la rotation en orbite. Cette méthode permet non seulement l'inspection d'objet mais aussi une exploration globale de l'environnement.

Dans le contexte d'inspection assistée d'objet, et de manière différente de la métaphore "*monde en main*", c'est la caméra et non l'objet qui se déplace. [Khan 2005] ont proposé une méthode *Hovercam* qui prend en compte les translations en plus de la rotation en maintenant une distance fixe entre la caméra et l'objet (Figure 2.11).

Dans le cadre des méthodes assistées, on retrouve celles où l'utilisateur désigne des positions dans l'environnement à visiter. L'utilisateur détermine la position d'intérêt et la caméra se déplace en direction de cette position d'une façon logarithmique : la vitesse de la caméra diminue au fur et à mesure qu'elle s'approche de la position d'intérêt [Mackinlay 1990]. Cependant, en raison des propriétés du logarithme, quand la distance entre la caméra et la position à atteindre est très grande la vitesse du mouvement de la caméra est très élevée. Ainsi, visuellement la trajectoire générée n'est pas toujours dynamiquement pertinente. Concernant le champ de vue, il est automatiquement orienté vers la surface à proximité du point d'intérêt durant la navigation en utilisant la normale à cette surface. D'autres techniques ont émergé s'inspirant de ce principe. Elles sont compatibles avec plusieurs systèmes d'affichage comme *Navidget* [Hachet 2008] (Figure 2.12) qui vise à explorer un large environnement sur une application mobile en définissant des positions à atteindre tout en considérant leur profondeurs.

Une extension de la méthode de *Mackinlay et al.* a été proposée par [Igarashi 1998],

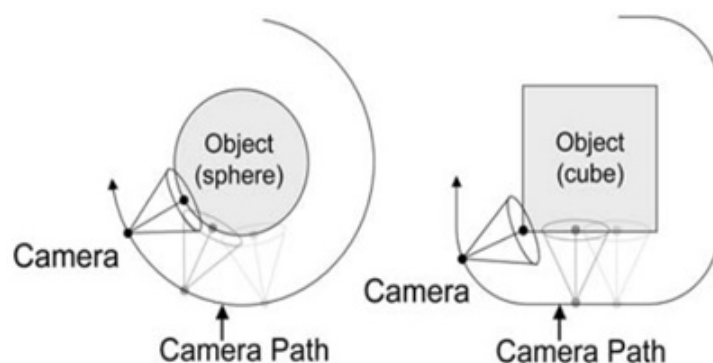


FIGURE 2.11 – Les Types de mouvement dans *Hovercam* [Khan 2005].

mais cette fois-ci, au lieu de choisir une position d'intérêt à atteindre, l'utilisateur dessine avec un trait la trajectoire (en 2D) qu'il veut réaliser et la caméra virtuelle suit cette trajectoire. Une approche similaire (Figure 2.13) mais plus avancée a été présentée par [Hagedorn 2008]. Ce système interprète automatiquement les traits dessinés par l'utilisateur selon leur géométrie, leur contexte spatial et temporel.

En ce qui nous concerne, nous proposerons une méthode qui permet à l'utilisateur de définir les positions d'intérêts à visiter. Cependant, la trajectoire liant la position courante de la caméra virtuelle et la position à atteindre est calculée d'une manière automatique en respectant des critères que nous avons prédéfini (Chapitre 4).

2.3.2 Méthodes automatiques

Les méthodes automatiques de contrôle de caméra permettent de générer automatiquement les trajectoires de la caméra dans un environnement 3D en exploitant des informations géométriques ou photométriques. Nous distinguons les méthodes de planification de trajectoire et les méthodes basées images.

2.3.2.1 Méthodes de planification de trajectoire

Dans les méthodes de planification de trajectoire, un ensemble de points de vue de la caméra est au préalable défini, soit manuellement par l'utilisateur, soit automatiquement en se basant sur certains critères de pertinence de points de vue comme présentés dans la section 2.2. Ensuite, des méthodes d'interpolation linéaire comme les splines [Astley 1994], associées à l'algorithme A^* [Björnsson 2005] peuvent être utilisées pour générer la trajectoire passant par ces points de vue. Pour la planification de trajectoire dans des environnements complexes, il existe des méthodes plus

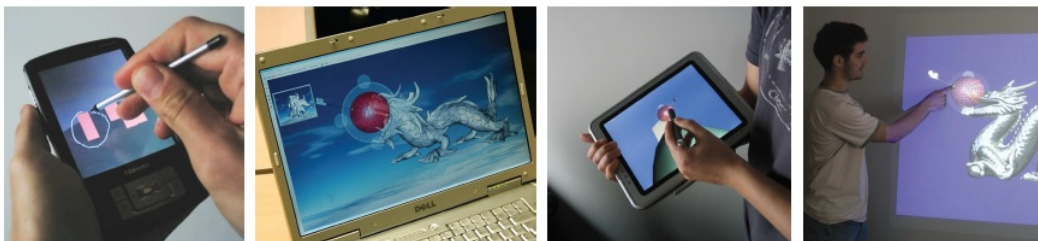


FIGURE 2.12 – Illustration de *Navidget* basé sur la technique des positions d'intérêts sur plusieurs systèmes d'affichage [Hachet 2008].

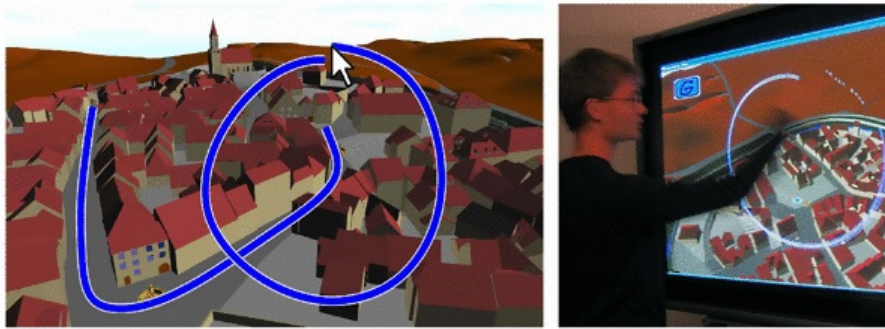


FIGURE 2.13 – Exploration d'une ville à l'aide de la méthode [Hagedorn 2008], l'utilisateur dessine la trajectoire que doit suivre la caméra.

appropriées comme les approches de "feuilles de route", les approches de décomposition en cellules et les approches à champs de potentiel.

Les approches de feuilles de route [Salomon 2003] [T. Li 2008] [Sun 2005] [Bhattacharya 2006] consistent à établir toutes les configurations existantes dans un espace libre. Ensuite, un graphe de connexions est construit, permettant de lier ces configurations. Ainsi, la feuille de route permet de lier une configuration initiale à une configuration désirée. Le graphe de visibilité [Nilsson 1969], le diagramme de Voronoï [Sharir 1983] et les méthodes basées silhouettes [Canny 1988] sont des exemples de "feuille de route". Dans [T. Li 2008], une méthode de planification de trajectoire à feuilles de route (Figure 2.14) a été développée afin de suivre un avatar et modifier sa position dans le but de lui fournir une meilleure vue de l'environnement. La particularité de cette méthode est que la feuille de route est modifiée en temps réel en fonction de la position de l'avatar. Les méthodes de "feuille de route" ont été appliquées aussi afin de générer des trajectoires tout en évitant les collisions [Li 1999].

Les méthodes de décomposition en cellules [Bandi 1998, Lamarche 2009] permettent de décomposer l'espace libre en cellules. Ensuite, un chemin connectant les configurations initiale et désirée est construit en fonction des points de vue de chaque cellule successive. [Andújar 2004] ont proposé dans ce contexte, une approche basée sur la décomposition en cellules pour la navigation dans un environnement 3D (Figure 2.15). Une fois l'environnement décomposé en cellules, la sélection des points de vue à visiter dans chaque cellule est réalisée en se basant sur un critère d'entropie géométrique. La trajectoire est générée de manière à relier ces points de vue.

Dans le cadre des méthodes de planification de trajectoire, on retrouve aussi les méthodes à champs de potentiel qui s'inspirent des théories de la physique, comme

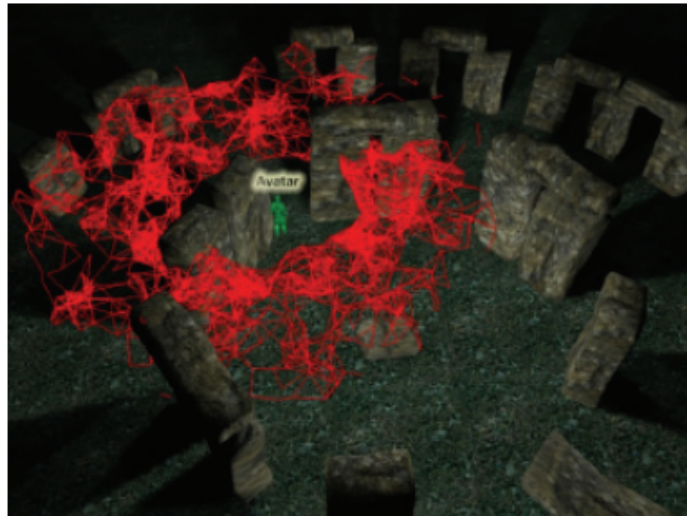


FIGURE 2.14 – Déplacement de l'avatar à l'aide du calcul de la "feuille de route" [T. Li 2008].

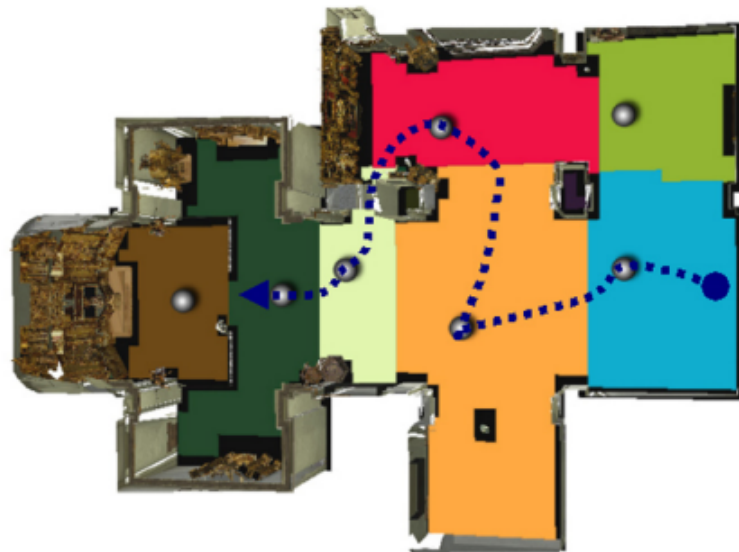


FIGURE 2.15 – Génération automatique d'une trajectoire passant par toutes les cellules [Andújar 2004].

celles concernant l'interaction des particules de charge positive ou négative dans un champ électrostatique. Ce type de méthode a été initialement utilisé dans le cadre de l'évitement d'obstacles en robotique [Khatib 1986]. Le problème de planification de trajectoire est modélisé en considérant l'obstacle comme une charge positive et

la cible (configuration désirée) par une charge négative. La trajectoire est calculée en appliquant une descente de gradient. Cette méthode a été exploitée pour réaliser une navigation automatique au sein du côlon [Paik 1998]. Le mouvement est calculé par un algorithme basé sur les champs de potentiel et le milieu du côlon a été modélisé comme un chemin dans le champ. Une approche à champs de potentiels [Beckhaus 2000] pour l'aide à l'exploration dans un environnement 3D a été proposée. Dans cette méthode, l'environnement est représenté par une grille rectangulaire uniforme et le système permet de générer automatiquement la trajectoire de la caméra en considérant tous les objets de la scène comme étant des obstacles.

2.3.2.2 Méthodes basées image

Les méthodes basées image sont inspirées du formalisme d'asservissement visuel. Elles permettent de générer les positions et les orientations de la caméra en exploitant des primitives de l'image. Dans [Gleicher 1992], une technique de contrôle de caméra basée asservissement visuel a été présentée. La vitesse de la caméra est calculée à l'aide d'un algorithme d'optimisation en satisfaisant des contraintes définies dans l'image. Ces contraintes sont modélisées à l'aide de primitives géométriques de l'image (position de deux points dans l'image par exemple). En utilisant ces mêmes principes, Courty *et al.* ont proposé une méthode de cadrage automatique d'un tableau dans un environnement 3D en asservissant la caméra sur des primitives géométriques [Courty 2001] (les quatre côtés du tableau) (Figure 2.16).

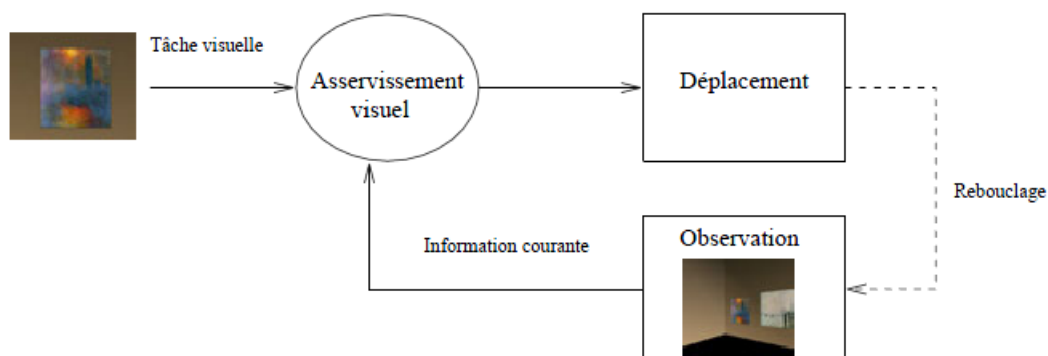


FIGURE 2.16 – Cadrage du tableau en se basant sur des primitives de l'image [Courty 2001].

En outre, dans le cadre de la navigation dans un environnement virtuel (un musée), ils ont défini une tâche principale [Courty 2001] qui consiste à suivre une sphère

dont la trajectoire a été prédéfinie au préalable en combinant des contraintes d'évitement d'obstacles et d'occultations. En utilisant cette fois des primitives photométriques [Marchand 2007] ont proposé une approche basée asservissement visuel pour la maximisation de la luminance. En effet, le but est d'atteindre le point de vue de caméra à partir duquel l'objet considéré est le mieux éclairé.

2.4 Conclusion

Dans ce chapitre, nous avons proposé un état de l'art sur les techniques existantes pour le calcul d'un point de vue pertinent. Il en ressort que parmi toutes les méthodes existantes, rares sont celles adaptées aux nuages de points colorés. En effet, la majorité concerne les scènes représentées à l'aide d'un maillage et elles n'utilisent pas suffisamment l'information visuelle. Nous proposons dans la suite de ce manuscrit, de décrire nos contributions pour le cadrage et pour l'exploration. Pour la modélisation de la pertinence visuelle nous avons utilisé l'entropie de Shannon et un mélange de gaussiennes basées saillance. Pour le calcul de l'entropie et du mélange, nous nous basons sur l'information photométrique et non sur l'information géométrique, ce qui rend notre modélisation plus générale et plus appropriée à tout type de scène 3D (maillage, nuage de points).

D'une autre part, nous avons présenté quelques méthodes de contrôle de caméra. En particulier, nous avons distingué les techniques purement manuelles, de celles assistées ou automatiques. Cependant, concernant les méthodes manuelles, la tâche de l'utilisateur face à ces techniques, qui ne nécessitent aucun pré-traitement ni de système d'aide, est très difficile. Les méthodes assistées ont été introduites pour faciliter l'interaction dans les environnements virtuels surtout quand il s'agit d'environnements vastes et complexes. Ces méthodes combinent des méthodes purement interactives (manuelles) à d'autres automatiques qui sont réalisées en arrière plan afin de faciliter la manipulation des interfaces d'interaction. Ensuite, nous avons présenté les méthodes automatiques où la trajectoire de caméra est calculée dans un processus hors-ligne en prenant en compte certains critères (comme des points de passages dans les méthodes de planification de trajectoires ou bien des primitives basées image). Les méthodes basées image s'avèrent être très intéressantes car la caméra est contrôlée directement selon ce que l'on souhaite apercevoir dans l'image. Afin d'avoir une vision plus large sur les techniques de contrôle de caméra, on peut se référer à l'art fait par [Christie 2005].

Cadrage visuel

Sommaire

3.1	Introduction	45
3.2	Cadrage visuel automatique basé entropie photométrique	45
3.2.1	Définition statistique de l'entropie de Shannon	46
3.2.2	Modélisation du problème : Entropie photométrique	46
3.2.3	Résolution du problème	48
3.2.4	Résultat sur un nuage de points 3D dense et coloré	51
3.2.5	Limitations	52
3.3	Cadrage visuel automatique : mélange de gaussiennes basées saillance	55
3.3.1	Critère de la saillance visuelle	55
3.3.2	Le mélange de gaussiennes représentant l'image de saillance	58
3.3.3	Définition des paramètres des gaussiennes	59
3.3.4	Expression optimale du λ	60
3.3.5	Résolution du problème	62
3.3.6	Résultat de simulation : environnement 2D	64
3.3.7	Résultat de simulation : environnement 3D	65
3.3.8	Résultat sur un nuage de points dense et coloré	68
3.3.9	Résultat en environnement réel	73
3.4	Cadrage visuel assisté	76
3.4.1	Loi de commande assistée	77
3.4.2	Résultat de simulation	79
3.4.3	Résultat sur un nuage de points 3D	79
3.4.4	Évaluation du système	82
3.5	Conclusion	85

3.1 Introduction

Dans l'art visuel, le cadrage est lié au concept de la pertinence du point de vue et à l'action d'atteindre ce point de vue. Ainsi, à partir d'une pose initiale de la caméra virtuelle, le cadrage visuel consiste à atteindre une autre pose de caméra où l'image rendue est plus pertinente en terme d'information visuelle, que l'image rendue initiale.

La pertinence visuelle s'exprime en fonction du contenu de l'image. Pour ce faire, deux primitives de pertinence présentées dans l'état de l'art vont être considérées, notamment l'entropie mais cette fois-ci photométrique, et la saillance visuelle.

Comme le cadrage visuel implique un mouvement de la caméra, la pertinence visuelle doit être liée aux d.d.l de la caméra. Il s'avère que l'asservissement visuel basé image est un formalisme qui permet de traiter cette problématique. Ce processus itératif met à jour la pose de la caméra, en optimisant un critère défini dans le plan image. L'utilisation directe de l'image de saillance comme primitive visuelle pour l'asservissement ne permet cependant pas d'atteindre un cadrage pertinent. Ceci va être expliqué plus en détail dans la section 3.3. Nous proposons alors un nouveau critère de pertinence qui est à la fois adapté à la pertinence et au contrôle de la caméra : une nouvelle modélisation de l'image de saillance par un mélange de gaussiennes. Ceci représente l'une de nos contributions clés, décrite dans la troisième section de ce chapitre. D'autres contraintes vont être ajoutées, tel que l'évitement d'obstacles, l'orientation réaliste de la caméra et la prise en compte de l'interaction de l'utilisateur dans le cas du cadrage assisté présenté dans la section 3.4.

Definition 1. *Le cadrage visuel est défini comme l'action d'atteindre un point de vue pertinent. A partir de ce point de vue, l'information pertinente ou saillante doit être centrée dans l'image et observée à partir du meilleur angle de vue.*

3.2 Cadrage visuel automatique basé entropie photométrique

Dans cette section, il est question de présenter le formalisme adopté pour réaliser le cadrage visuel en exploitant l'entropie photométrique comme primitive de pertinence visuelle. Il est donc nécessaire d'introduire l'entropie photométrique et de décrire le processus de résolution. Quelques résultats du succès de l'approche sont présentés en expliquant ses limites ensuite.

3.2.1 Définition statistique de l'entropie de Shannon

D'un point de vue statistique, *Claude Shannon* [Shannon 1948] a introduit la quantité d'information d'une variable aléatoire en fonction de sa variation. L'entropie de Shannon $E(X)$ permet de mesurer la quantité d'information d'une variable aléatoire discrète X tel que $X \in \Omega_X = \{x_1, \dots, x_n\}$:

$$E(X) = - \sum_{1 \leq i \leq n} P_i \log_2(P_i) \quad (3.1)$$

où la probabilité $P_i = P(X = x_i)$ avec $x_i \in \Omega_X$. La base du logarithme est égale à 2 car l'entropie est exprimée en bits. Plus la valeur de P_i est élevée, plus c'est évident que $X = x_i$ et plus P_i est faible, plus c'est improbable que $X = x_i$. D'un autre côté, plus P_i est élevée, plus $-\log_2(P_i)$ est faible et plus P_i est faible, plus $-\log_2(P_i)$ est élevé. Ainsi $-\log_2(P_i)$ est une mesure d'incertitude de la variable x_i ce qui représente l'information associée à x_i .

3.2.2 Modélisation du problème : Entropie photométrique

On considère $I(\mathbf{r})$ l'image rendue à partir de la pose $\mathbf{r} = (t_X, t_Y, t_Z, \theta w_X, \theta w_Y, \theta w_Z)^\top$, où (t_X, t_Y, t_Z) sont les trois translations selon les axes \vec{X} , \vec{Y} et \vec{Z} , respectivement. Les rotations sont représentées par un angle θ et un vecteur unitaire (axe de rotation) $\mathbf{w} = [w_X, w_Y, w_Z]^\top$. L'entropie photométrique d'une image est donnée par :

$$E(\mathbf{r}) = - \sum_i P_i(\mathbf{r}) \log_2(P_i(\mathbf{r})) \quad (3.2)$$

tel que i est un niveau de gris d'un pixel dans l'image, $i \in [0; 255]$ et $P_i(\mathbf{r})$ la probabilité que l'intensité i existe dans l'image I . Pour une meilleure lisibilité $P_i(\mathbf{r})$ est remplacée par P_i dans ce qui suit.

La distribution de probabilité P_i est obtenue en utilisant l'histogramme d'intensités normalisé par le nombre total des pixels de l'image h :

$$P_i = \frac{1}{N_{\mathbf{u}}} h(i, \mathbf{r}) \quad (3.3)$$

qu'on peut exprimer en fonction de l'image I comme suit :

$$P_i = \frac{1}{N_{\mathbf{u}}} \sum_{\mathbf{u}} \delta_0(i - I(\mathbf{u}, r)) \quad (3.4)$$

Où $I(\mathbf{u}, \mathbf{r})$ est l'intensité au pixel $\mathbf{u} = (u, v)$, $N_{\mathbf{u}}$ est le nombre de pixels de l'image et $\delta(x)$ représente la fonction Kronecker :

$$\delta(x) = \begin{cases} 1 & \text{si } x = 0 \\ 0 & \text{sinon} \end{cases} \quad (3.5)$$

Cette fonction imite le principe du calcul d'un histogramme. En effet, une classe de l'histogramme est incrémentée à chaque fois qu'une intensité d'un pixel de l'image est comprise dans cette classe. Ainsi, l'entropie représente la variabilité dans l'image, qui peut être considérée comme une mesure de l'équilibre de son l'histogramme.

Dans un contexte de recalage par information mutuelle entre deux images, l'histogramme normalisé est approximé à l'aide d'une B-Spline d'ordre 4, notée ϕ [Dame 2009] : "ceci permet de lisser et de rendre la fonction d'entropie différentiable". Le calcul des valeurs d'une B-Spline et de ses dérivées est simple, d'où l'avantage de son utilisation dans notre problème. Effectivement, une optimisation du second ordre est utilisée afin de résoudre le problème du cadrage visuel.

La théorie des B-Spline introduite par Schoenberg en 1946 joue un rôle très important dans les méthodes d'approximation. Une B-Spline cardinale du premier ordre notée par ϕ_1 est une fonction caractéristique de l'intervalle $[-0.5, 0.5[$,

$$\phi_1(i) = \begin{cases} 1 & \text{si } i \in [-0.5, 0.5[\\ 0 & \text{sinon} \end{cases} \quad (3.6)$$

Une généralisation de la définition d'une B-spline d'ordre n est donnée par la relation récursive suivante :

$$\phi_n(i) = (\phi_n * \phi_1)(i) \quad (3.7)$$

tel que $*$ représente le produit de convolution.

Les dérivées d'une B-Spline se calculent à l'aide des B-Spline d'ordre inférieur :

$$\frac{\partial(\phi_n(i))}{\partial i} = \phi_{n-1}(i + \frac{1}{2}) - \phi_{n-1}(i - \frac{1}{2}) \quad (3.8)$$

Les B-Splines d'ordre 2, 3 et 4 sont données respectivement par :

$$\phi_2(i) = \begin{cases} 1 + i & \text{si } i \in [-1, 0[\\ 1 - i & \text{si } i \in [0, 1[\\ 0 & \text{sinon} \end{cases} \quad (3.9)$$

$$\phi_3(i) = \begin{cases} (1.5 + i)^2 / 2 & \text{si } i \in [-1.5, -0.5[\\ 1 + i - (0.5 + i)^2 & \text{si } i \in [-0.5, 0[\\ 1 - i - (0.5 - i)^2 & \text{si } i \in [0, 0.5[\\ (1.5 - i)^2 / 2 & \text{si } i \in [0.5, 1.5[\\ 0 & \text{sinon} \end{cases} \quad (3.10)$$

$$\phi_4(i) = \begin{cases} ((2 + i)^3) / 6 & \text{si } i \in [-2, -1[\\ (1 + 3(1 + i) + 3(1 + i)^2 - 3(1 + i)^3) / 6 & \text{si } i \in [-1, 0[\\ (1 + 3(1 - i) + 3(1 - i)^2 - 3(1 - i)^3) / 6 & \text{si } i \in [0, 1[\\ ((2 - i)^3) / 6 & \text{si } i \in [1, 2[\\ 0 & \text{sinon} \end{cases} \quad (3.11)$$

En approximant l'histogramme normalisé par une B-Spline, l'expression de la distribution de probabilité est donc donnée par :

$$P_i = \frac{1}{N_{\mathbf{u}}} \sum_{\mathbf{u}} \phi(i - I(\mathbf{u}, \mathbf{r})) \quad (3.12)$$

3.2.3 Résolution du problème

La fonction caractérisant la pertinence, c'est-à-dire l'entropie photométrique, est non linéaire par rapport à la pose de la caméra \mathbf{r} . Un processus de résolution non linéaire itératif est appliqué. La linéarisation est effectuée à l'aide du développement de Taylor. Le processus itératif consiste à estimer l'incrément de pose $\dot{\mathbf{r}}_k$ afin de modifier la pose \mathbf{r}_{k-1} (itération $k - 1$) en utilisant l'exponentielle map $e^{[\dot{\mathbf{r}}_k]}$:

$$\mathbf{r}_k = e^{[\dot{\mathbf{r}}_k]} \mathbf{r}_{k-1} \quad (3.13)$$

3.2.3.1 Critère de l'entropie photométrique

La maximisation de l'entropie photométrique est équivalente à la minimisation de son opposé $-E(\mathbf{r}_k)$. Ainsi, la recherche du maximum revient à calculer les racines de la dérivée de $-E(\mathbf{r}_k)$. Plusieurs algorithmes existent pour résoudre ce problème comme nous avons vu dans le Chapitre 1. L'algorithme du second ordre de Newton est retenu car il est plus robuste vis-à-vis des optimums locaux comparé à ceux du premier ordre.

Pour maximiser l'entropie photométrique, on cherche à annuler la Jacobienne suivante :

$$\frac{\partial (-E(\mathbf{r}_k))}{\partial \mathbf{r}} = 0 \quad (3.14)$$

Notons $d_{\mathbf{r}}(-E(\mathbf{r}_k))$ la dérivée partielle de la fonction $-E(\mathbf{r}_k)$ par rapport à \mathbf{r} , et $d_{\mathbf{r}}^2(-E(\mathbf{r}_k))$ la dérivée partielle d'ordre 2. Le développement de Taylor du 1^{er} ordre est appliqué en \mathbf{r}_k :

$$d_{\mathbf{r}}(-E(\mathbf{r}_k)) \simeq d_{\mathbf{r}}(-E(\mathbf{r}_{k-1})) + d_{\mathbf{r}}^2(-E(\mathbf{r}_{k-1}))\dot{\mathbf{r}}_k = 0 \quad (3.15)$$

Pour résoudre cette équation, la Jacobienne et le Hessian de la fonction de coût doivent être calculés. En utilisant l'équation (3.2) la Jacobienne de l'entropie photométrique est donnée par :

$$\begin{aligned} d_{\mathbf{r}}(-E(\mathbf{r}_{k-1})) &= \frac{\partial}{\partial \mathbf{r}} \left(\sum_i P_i \log_2(P_i) \right) \\ &= \sum_i \frac{\partial P_i}{\partial \mathbf{r}} \log_2(P_i) + P_i \frac{\partial P_i}{\partial \mathbf{r}} \frac{1}{P_i \log(2)} \\ &= \sum_i \frac{\partial P_i}{\partial \mathbf{r}} \left(\frac{1}{\log(2)} + \log_2(P_i) \right) \end{aligned} \quad (3.16)$$

En utilisant l'expression de la probabilité exprimée dans l'équation (3.12), on obtient :

$$\frac{\partial P_i}{\partial \mathbf{r}} = \frac{\partial}{\partial \mathbf{r}} \left(\frac{1}{N_{\mathbf{u}}} \sum_{\mathbf{u}} \phi(i - I(\mathbf{u}, \mathbf{r}_{k-1})) \right) \quad (3.17)$$

Ainsi,

$$\frac{\partial P_i}{\partial \mathbf{r}} = \frac{1}{N_{\mathbf{u}}} \sum_{\mathbf{u}} \frac{\partial \phi(i - I(\mathbf{u}, \mathbf{r}_{k-1}))}{\partial \mathbf{r}} \quad (3.18)$$

On sait que $\frac{d}{dx} (g \circ f(x)) = \frac{dg(f(x))}{df(x)} \frac{df(x)}{dx}$. On a donc :

$$\begin{aligned} \frac{\partial \phi(i - I(\mathbf{u}, \mathbf{r}_{k-1}))}{\partial \mathbf{r}} &= \frac{\partial \phi(i - I(\mathbf{u}, \mathbf{r}_{k-1}))}{\partial (i - I(\mathbf{u}, \mathbf{r}_{k-1}))} \frac{\partial (i - I(\mathbf{u}, \mathbf{r}_{k-1}))}{\partial \mathbf{r}} \\ &= \frac{\partial \phi(i - I(\mathbf{u}, \mathbf{r}_{k-1}))}{\partial (i - I(\mathbf{u}, \mathbf{r}_{k-1}))} \frac{\partial i}{\partial \mathbf{r}} - \frac{\partial \phi(i - I(\mathbf{u}, \mathbf{r}_{k-1}))}{\partial (i - I(\mathbf{u}, \mathbf{r}_{k-1}))} \frac{\partial I(\mathbf{u}, \mathbf{r}_{k-1})}{\partial \mathbf{r}} \end{aligned} \quad (3.19)$$

avec $\frac{\partial i}{\partial \mathbf{r}} = 0$ car i représente le nombre de classes de l'histogramme qui est fixe. Ainsi, on aura :

$$\begin{aligned} \frac{\partial \phi(i - I(\mathbf{u}, \mathbf{r}_{k-1}))}{\partial \mathbf{r}} &= - \frac{\partial \phi(i - I(\mathbf{u}, \mathbf{r}_{k-1}))}{\partial(i - I(\mathbf{u}, \mathbf{r}_{k-1}))} \frac{\partial I(\mathbf{u}, \mathbf{r}_{k-1})}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{r}} \\ &= - \frac{\partial \phi(i - I(\mathbf{u}, \mathbf{r}_{k-1}))}{\partial(i - I(\mathbf{u}, \mathbf{r}_{k-1}))} \nabla I \mathbf{L}_{\mathbf{u}} \end{aligned} \quad (3.20)$$

La dérivée de la distribution de probabilité est finalement donnée par :

$$\frac{\partial P_i}{\partial \mathbf{r}} = \frac{1}{N_{\mathbf{u}}} \sum_{\mathbf{u}} \left(- \frac{\partial \phi(i - I(\mathbf{u}, \mathbf{r}_{k-1}))}{\partial(i - I(\mathbf{u}, \mathbf{r}_{k-1}))} \nabla I \mathbf{L}_{\mathbf{u}} \right) \quad (3.21)$$

où $\nabla I = (\frac{\partial I}{\partial u}, \frac{\partial I}{\partial v})$ est le gradient de l'image $I(\mathbf{r})$ et $\mathbf{L}_{\mathbf{u}}$ la matrice d'interaction. Elle lie les variations dans l'image aux variations de la pose de la caméra au pixel $\mathbf{u} = (u, v)$, et se définit par :

$$\mathbf{L}_{\mathbf{u}} = \frac{\partial \mathbf{u}}{\partial \mathbf{x}} \mathbf{L}_{\mathbf{x}} \quad (3.22)$$

avec $\mathbf{L}_{\mathbf{x}}$ la matrice définie par l'équation (1.22).

Le Hessian de l'entropie photométrique est donné par :

$$d_{\mathbf{r}}^2(-E(\mathbf{r}_{k-1})) = \sum_i \frac{\partial P_i}{\partial \mathbf{r}} \left(\frac{\partial P_i}{\partial \mathbf{r}} \right)^T \frac{1}{P_i} + \frac{\partial^2 P_i}{\partial \mathbf{r}^2} (1 + \log(P_i)) \quad (3.23)$$

En utilisant la première dérivée de probabilité $\frac{\partial P_i}{\partial \mathbf{r}}$, on calcule sa dérivée seconde, ce qui donne :

$$\frac{\partial^2 P_i}{\partial \mathbf{r}^2} = \frac{1}{N_{\mathbf{u}}} \sum_{\mathbf{u}} \left(\frac{\partial^2 \phi(i - I(\mathbf{u}, \mathbf{r}_{k-1}))}{\partial(i - I(\mathbf{u}, \mathbf{r}_{k-1}))^2} \mathbf{L}_I^T \mathbf{L}_I - \frac{\partial \phi(i - I(\mathbf{u}, \mathbf{r}_{k-1}))}{\partial(i - I(\mathbf{u}, \mathbf{r}_{k-1}))} \mathbf{H}_I \right) \quad (3.24)$$

On suppose que la scène est lambertienne, condition vérifiée pour les petits déplacements. La matrice d'interaction \mathbf{L}_I et la Hessienne \mathbf{H}_I de l'intensité d'un pixel sont données en utilisant la définition de [Collewet 2008] par :

$$\mathbf{L}_I = \nabla I \mathbf{L}_{\mathbf{u}} \quad (3.25)$$

$$\mathbf{H}_I = \mathbf{L}_{\mathbf{u}}^T \nabla^2 I \mathbf{L}_{\mathbf{u}} + \frac{\partial I}{\partial u} \mathbf{H}_u + \frac{\partial I}{\partial v} \mathbf{H}_v \quad (3.26)$$

tel que $\nabla^2 I = \begin{pmatrix} \frac{\partial^2 I}{\partial u^2} & \frac{\partial^2 I}{\partial u \partial v} \\ \frac{\partial^2 I}{\partial v \partial u} & \frac{\partial^2 I}{\partial v^2} \end{pmatrix}$ est le gradient du gradient de l'image. \mathbf{H}_u et \mathbf{H}_v sont les Hessiens de taille 6×6 des deux coordonnées du point \mathbf{u} [Lapresté 2004] :

$$H_u = \begin{pmatrix} 0 & 0 & -V^2 & 0 & 2uV & -vV \\ 0 & 0 & 0 & -V & 0 & -uV \\ -V^2 & 0 & 2uV^2 & vV & -2u^2V & 2uvV \\ 0 & -V & vV & -u & -2uv & v^2 - u^2 \\ 2uV & 0 & -2u^2V & -2uv & 2u(1 + u^2) & -v(1 + 2u^2) \\ -vV & -uV & 2uvV & v^2 - u^2 & -v(1 + 2u^2) & u(2v^2 + 1) \end{pmatrix} \quad (3.27)$$

$$H_v = \begin{pmatrix} 0 & 0 & 0 & V & vV & 0 \\ 0 & 0 & -V^2 & 0 & uV & -2vV \\ 0 & -V^2 & 2vV^2 & -uV & -2uvV & 2v^2V \\ V & 0 & -uV & -v & u^2 - v^2 & -2uv \\ vV & uV & -2uvV & u^2 - v^2 & v(2u^2 + 1) & -2uv^2 \\ 0 & -2vV & 2v^2V & -2uv & -2uv^2 & 2v(1 + v^2) \end{pmatrix} \quad (3.28)$$

où $V = 1/Z$, avec Z la profondeur du point.

Ainsi, la loi de commande permettant de maximiser l'entropie photométrique dans l'image est calculée à partir de la Jacobienne et du Hessien :

$$\dot{\mathbf{r}}_k = \beta (d_{\mathbf{r}}^2(-E(\mathbf{r}_{k-1})))^+ d_{\mathbf{r}}(-E(\mathbf{r}_{k-1})) \quad (3.29)$$

où β est un scalaire de gain positif qui permet de régler la vitesse de convergence.

3.2.4 Résultat sur un nuage de points 3D dense et coloré

L'approche proposée est validée sur un modèle 3D simple constitué de nuages de points colorés. L'environnement est obtenu par relevés laser (Faro Focus 3D). Pour la visualisation du modèle, le moteur graphique Ogre 3D [Ogr], basé sur OpenGL a été utilisé.

Dans cette expérience, le comportement de la maximisation de l'entropie photométrique est mis en avant. Cette loi de commande est appliquée sur un modèle simple : un mur blanc et un poster accroché. En appliquant la maximisation de l'entropie photométrique, on remarque que le poster devient de plus en plus présent dans l'image au fur et à mesure du mouvement de la caméra (Figure 3.1). Ceci représente un résultat qui valide l'utilisation de l'entropie photométrique comme primitive visuelle permettant de maximiser l'information dans l'image lorsque l'histogramme des niveaux de gris est équilibré. Cependant l'utilisation de l'entropie

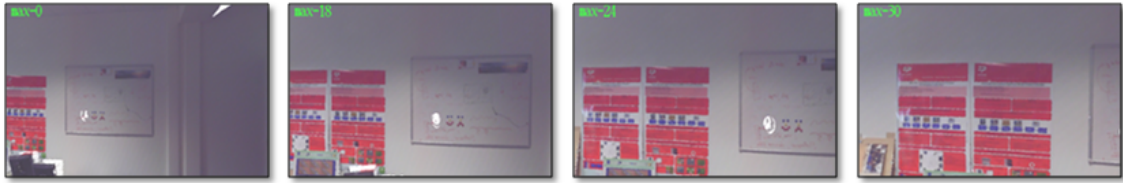


FIGURE 3.1 – La trajectoire visuelle de la caméra en maximisant la quantité d’information pertinente dans l’image.

photométrique pour le cadrage n’est pas appropriée, dans la mesure où la définition (Definition 1) du cadrage est à moitié respectée. En effet, l’information de pertinence est dispersée dans l’image mais non centrée.

La Figure 3.2 montre l’évolution de l’entropie photométrique relative au cadrage du poster dans l’image. Comme on peut le voir sur la courbe, l’entropie photométrique croît au fur et à mesure que le poster est inclus dans le champ de vue de la caméra.

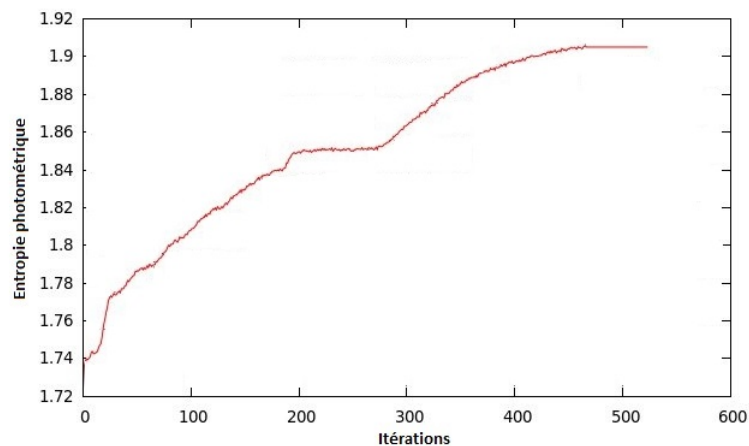


FIGURE 3.2 – Évolution de l’entropie photométrique.

3.2.5 Limitations

Pour mieux illustrer les limitations, deux objets de synthèse sont considérés, l’un bidimensionnel et l’autre tridimensionnel.

3.2.5.1 Résultat de simulation : objet 2D

On applique notre méthode de cadrage visuel basé entropie photométrique dans un environnement 2D constitué du plan d'une image. Le but de cette expérience est de voir si la maximisation de l'entropie photométrique sur deux d.d.l de la caméra, la translation t_X selon l'axe \vec{X} et la translation t_Y selon l'axe \vec{Y} , permet de cadrer l'information, c'est-à-dire le croquis du visage de la figure 3.3.

Dans cette simulation, le cadre initial est délimité par un rectangle blanc en pointillé et le cadre final par un rectangle blanc en un trait continu (le cadre initial n'est pas visible sur la figure car il est caché par le cadre final). L'évolution de la fonction de coût basée entropie photométrique (courbe en bas à droite de la figure 3.3) et les deux translations sont représentées. Quand la loi de commande basée maximisation de l'entropie photométrique est appliquée, on remarque que la caméra virtuelle reste bloquée sur le cadre initial (position initiale) comme le montre la figure 3.3. Ainsi, on observe sur ce résultat que le cadrage visuel n'est pas pertinent, c'est-à-dire que le visage n'est pas inclus en sa totalité dans toute l'image.

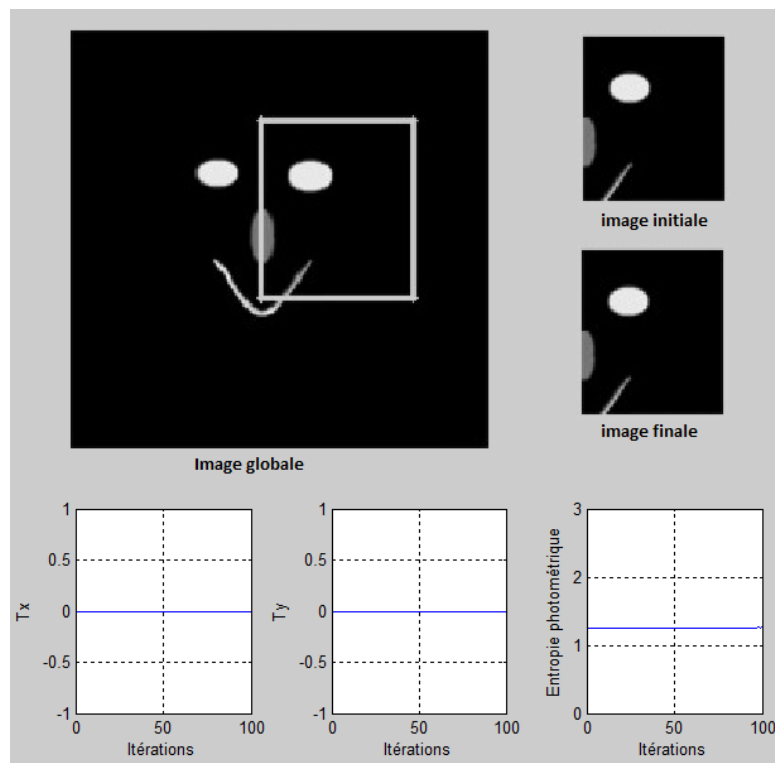


FIGURE 3.3 – Résultat de cadrage en maximisant l'entropie photométrique.

Le calcul de l'entropie repose sur l'histogramme des niveaux de gris dans l'image. Si l'histogramme n'est pas équilibré (dans le cas de l'utilisation du visage, plusieurs classes de l'histogramme sont nulles même en diminuant le nombre de classes à 8), l'incrément de pose obtenu est très faible. Même si on augmente considérablement le gain β de la loi de commande, les déplacements restent très faibles. Dans l'expérimentation suivante, on a choisi d'utiliser un objet texturé dans le but d'avoir différents niveaux de gris dans l'image utilisée pour le calcul de l'entropie photométrique.

3.2.5.2 Résultat de simulation : objet 3D

On considère un environnement 3D de synthèse simple, constitué d'une pièce avec un objet texturé. On applique la loi de commande (Équation (3.29)) dans le but de réaliser une navigation automatique en utilisant quatre d.d.l de la caméra $(t_X, t_Y, \theta_{w_X}, \theta_{w_Y})$. La translation sur l'axe \vec{Z} a été bloquée afin de garder une distance fixe entre la caméra et l'objet ainsi que la rotation autour de l'axe \vec{Z} , pour éviter les rotations irréalistes autour de l'axe optique de la caméra. D'après la séquence d'images clés de la figure 3.4, il est à remarquer que la caméra virtuelle s'est déplacée légèrement vers la droite pour inclure dans son champ de vue tout l'objet mais elle s'est arrêtée (l'algorithme arrive assez rapidement à un maximum local). La figure 3.5 montre que l'évolution de l'entropie photométrique est très faible et que les vitesses sont très proches de 0, ce qui explique l'arrêt du mouvement. Ainsi, le résultat de cadrage obtenu (Figure 3.4) n'est pas satisfaisant dans la mesure où l'objet n'est pas inclus dans sa totalité dans l'image finale et il n'est pas centré. Nous allons proposer, dans la section 3.3 une solution pour résoudre ce problème.

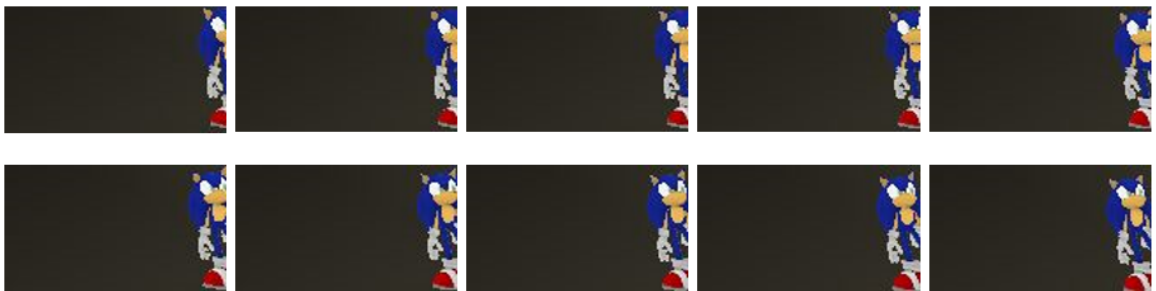


FIGURE 3.4 – Un échantillon de la séquence d'images obtenue en maximisant l'entropie photométrique.

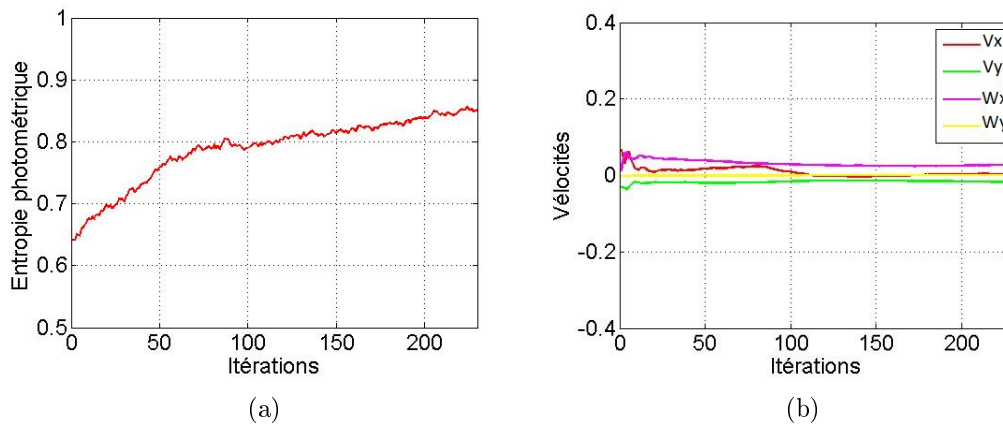


FIGURE 3.5 – (a) L'évolution de l'entropie photométrique ; (b) Les vitesses des 4 d.d.l de la caméra.

3.3 Cadrage visuel automatique : mélange de gaussiennes basées saillance

Un cadrage réussi dans notre contexte consiste à inclure l'information de pertinence dans l'image rendue et à la répartir d'une manière centrée sur toute l'image. Cette condition n'est pas toujours remplie par le cadrage basé entropie photométrique. En effet, il ne fonctionne correctement que sur des environnements 3D dont les images rendues ont un histogramme de niveaux gris équilibré. Par conséquent, une autre fonction de coût doit être envisagée. Cette section explore la saillance visuelle comme fonction de pertinence. [Radhakrishna 2009] et [Itti 1998] ont démontré que la saillance visuelle est une représentation adéquate de l'information de pertinence pour l'exploration visuelle humaine. De plus, ce critère (Chapitre 2) n'est pas dépendant de l'équilibre des niveaux de gris utilisés pour le calculer, contrairement à l'entropie photométrique.

3.3.1 Critère de la saillance visuelle

On commence par calculer la carte de saillance I_s de l'image à la pose \mathbf{r} . Ensuite, on définit la fonction de coût à considérer pour illustrer la pertinence visuelle par la somme des intensités de l'image de saillance afin de pouvoir contrôler la saillance de l'objet dans l'image. Ainsi, le critère à maximiser est donné par :

$$S(\mathbf{r}) = \sum_{N_j} I_s(\mathbf{r}) \quad (3.30)$$

où N_j est le nombre de pixels saillants dans l'image I_s .

Pour simplifier, on considère un signal mono dimensionnel I_s ne contenant qu'une seule valeur saillante à la position μ . Ce signal se modélise à l'aide d'une impulsion :

$$I_s(u) = \begin{cases} A & \text{si } u = \mu \\ 0 & \text{sinon} \end{cases} \quad (3.31)$$

où A est la valeur de saillance.

En considérant les deux signaux de la figure 3.6, la fonction de coût (Équation (3.30)) ne change pas malgré la translation du signal (de la caméra). Elle reste constante (égale à 0.58) quelque soit u de l'ensemble Ω , le domaine de définition de μ .

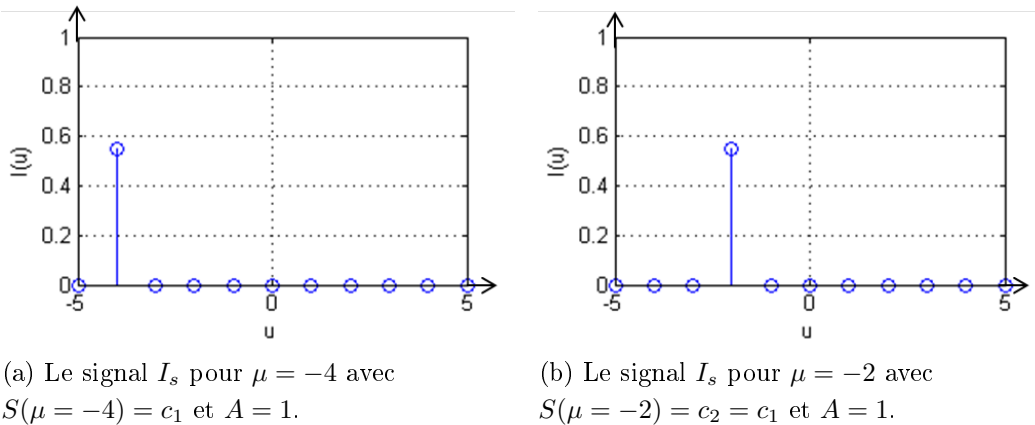


FIGURE 3.6 – Le signal avec une seule valeur saillante. La fonction de coût reste constante quelque soit la translation du signal dans le cadre.

Si on suppose que μ appartienne à Ω , alors S est égale à A . S est une fonction constante, la dérivée première et seconde sont égales à 0. Ainsi, l'incrément de translation sera toujours égal à 0, ce qui empêche la translation du signal. Par conséquent la saillance ne peut pas être utilisée directement pour le cadrage visuel.

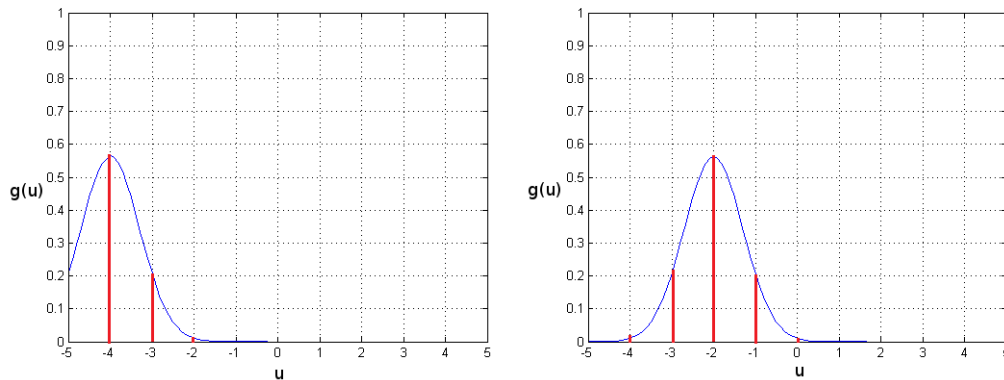
Nous proposons une solution à ce problème qui consiste à remplacer la fonction d'impulsion par une fonction gaussienne centrée en μ avec un écart type σ :

$$g(u, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(u-\mu)^2}{2\sigma^2}\right) \quad (3.32)$$

De ce fait, nous adaptons la fonction de coût de l'équation (3.30) :

$$G_1(r) = \sum_{u \in \Omega} g(u) \quad (3.33)$$

Cette fonction de coût, dont le domaine de définition est $]-\infty, \infty[$, représente la surface sous la gaussienne dans l'intervalle Ω . L'intérêt, c'est que cette surface change après une translation du signal (figure 3.7). Donc, en utilisant la définition de la fonction de coût $G_1(r)$, ses dérivées sont non nulles. De plus, d'après les deux figures 3.7a et 3.7b, on remarque que la surface sous la gaussienne est supérieure quand elle est mieux centrée. La gaussienne basée saillance apparaît donc plus intéressante pour réaliser le cadrage visuel.



(a) La fonction gaussienne centrée en $\mu = -4$ avec $G_1(\mu = -4) = c_1$.

(b) La fonction gaussienne centrée en $\mu = -2$ avec $G_1(\mu = -2) = c_2 > c_1$.

FIGURE 3.7 – La fonction gaussienne pour un signal avec une seule valeur de saillance.

En pratique, il n'y pas qu'une seule valeur saillante mais plusieurs. Considérons alors l'exemple d'un signal constitué de plusieurs valeurs de saillance. Pour illustrer le principe, on suppose qu'il contient trois valeurs. Chaque valeur saillante est modélisée par une gaussienne (Figure 3.8). Dans le but de prendre en compte toutes les valeurs de saillance dans le signal, nous proposons de représenter le signal à l'aide d'un mélange de gaussiennes. Un mélange de gaussiennes est la somme de composantes de densités de gaussiennes :

$$G(r) = \sum_u \sum_j g_j(u, \mu_j, \sigma_j) \quad (3.34)$$

où u est l'échantillon sur lequel est calculé le mélange et j est l'index des valeurs saillantes dans le signal. Le mélange de gaussiennes pour ce signal à trois valeurs

de saillance est représenté par la courbe en pointillés avec des marqueurs circulaires sur la figure 3.8.

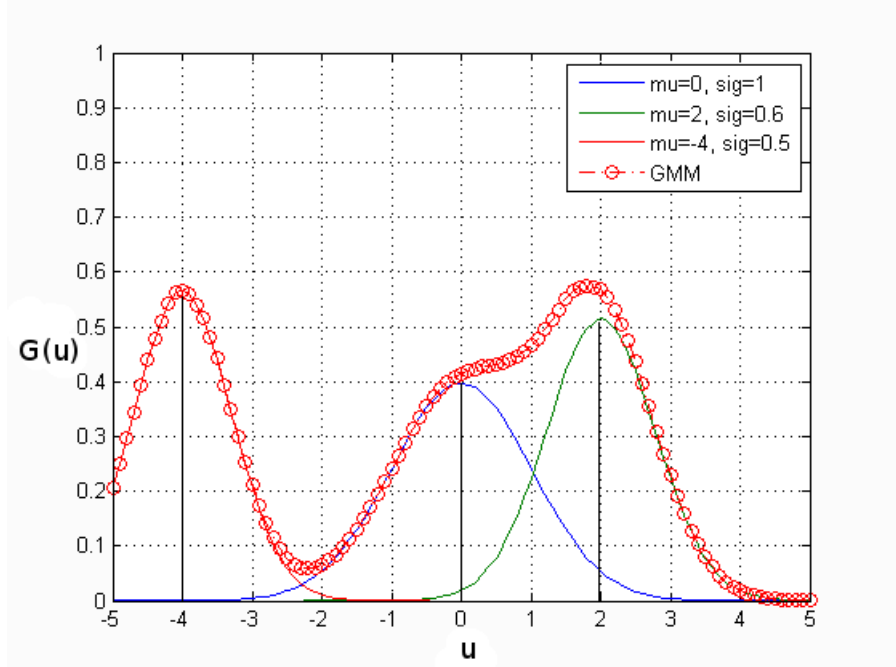


FIGURE 3.8 – La fonction du mélange de gaussiennes pour un signal constitué de trois valeurs saillantes.

La section 3.3.3 reviendra plus en détails sur les différents paramètres et notamment le σ .

3.3.2 Le mélange de gaussiennes représentant l'image de saillance

Chaque pixel d'indice j de l'image de saillance $I_s(\mathbf{r})$ est modélisé à l'aide d'une gaussienne centrée sur ce pixel ($\boldsymbol{\mu}_j = (u_j, v_j)$), ce qui a aussi pour effet d'éviter des traitements d'images complexes. L'image I_s est alors modélisée par un mélange de gaussiennes, tel que \mathbf{u} est l'échantillon utilisé pour calculer le mélange (tous les pixels de l'image). On étend l'équation (3.34) à deux dimensions :

$$G(\mathbf{r}) = \sum_{\mathbf{u}} \sum_j g(\mathbf{u}, \boldsymbol{\mu}_j, \sigma_j) \quad (3.35)$$

Tous les pixels de l'image sont utilisés pour le calcul du mélange de gaussiennes et font donc partie de cet échantillon. $g(\mathbf{u}, \boldsymbol{\mu}_j, \sigma_j)$ est une composante des densités

du mélange pour le pixel saillant d'index j (cet index ne parcourt que les pixels saillants de l'image). Notons que le mélange de gaussiennes est utilisé comme une nouvelle modélisation de l'image de saillance et non d'un point de vue statistique.

3.3.3 Définition des paramètres des gaussiennes

Pour distinguer les pixels saillants entre eux, on définit σ_j proportionnelle à l'intensité de l'image de saillance $I_s(\mu_j)$:

$$\sigma_j = \lambda I_s(\mu_j) \tag{3.36}$$

où le coefficient λ , strictement positif, est le même pour tous les pixels de l'image.

Considérons une image de taille 100×100 constituée d'un seul pixel saillant en blanc (Figure 3.9a).

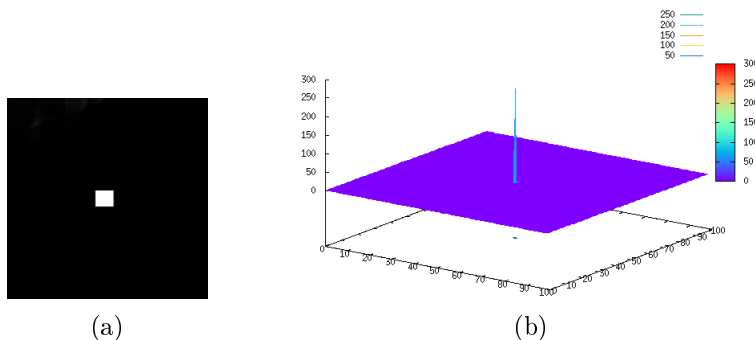


FIGURE 3.9 – (a)Image de saillance, (b) l'imeg sous forme d'impulsions.

Quand λ tend vers 0 (Figure 3.10a), le mélange de gaussiennes se rapproche de l'image initiale (Figure 3.9b). A contrario, plus il est grand, plus le mélange de gaussiennes est étendu (Figure 3.10b et Figure 3.10c). Ainsi, λ modifie la convexité du mélange et son expansion. On se rapproche donc du cas présenté sur la figure 3.7. Le volume sous la surface du mélange est maximal quand ce dernier est centré. Ainsi, λ a une influence sur la qualité du cadrage et aussi sur la convergence parce qu'il modifie la courbure de la fonction de coût. Par conséquent, plus λ est petit, plus σ_j est petit et moins le cadrage correspondra à la définition 1 (centrer l'information pertinente). Inversement, plus λ est grand, meilleur sera le cadrage. Mais, quand λ est trop grand (Figure 3.10d) le mode central du mélange de gaussiennes remplit toute la surface de l'image. Même si le mélange est différent de celui au λ faible, la fonction de coût devient plate ce qui empêche aussi un cadrage correct.

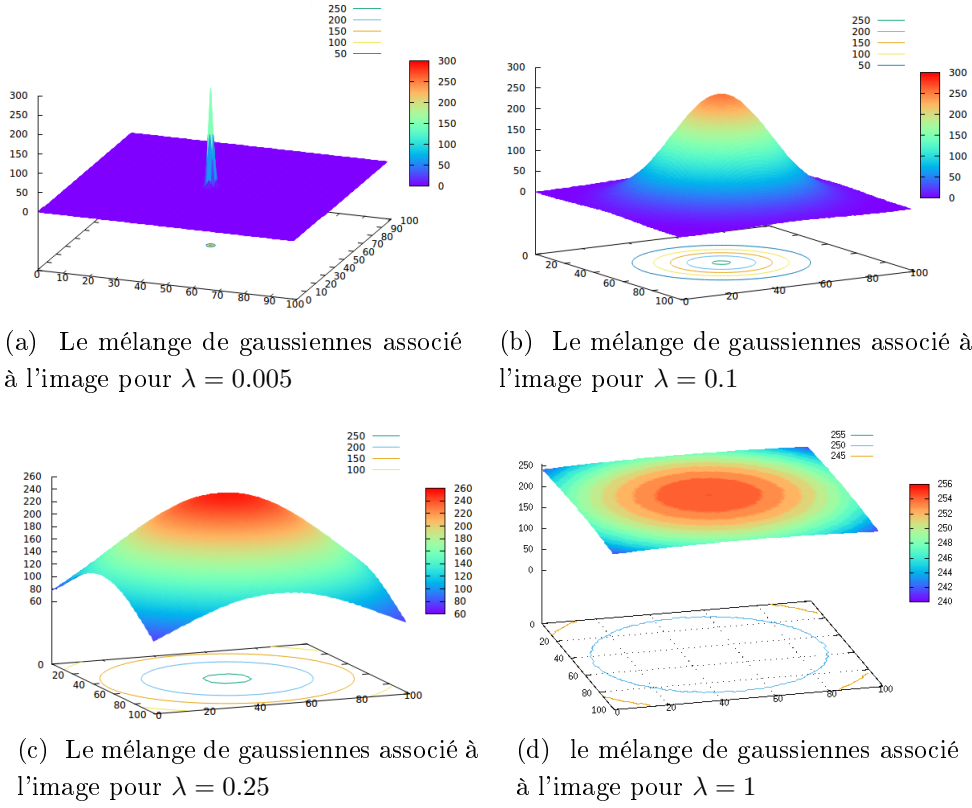


FIGURE 3.10 – Influence du λ sur le mélange de gaussiennes considérant une image simple avec un seul pixel saillant.

3.3.4 Expression optimale du λ

Pour éviter de définir le λ empiriquement, la condition de convexité a été exploitée. Par définition, une fonction est convexe si et seulement si sa dérivée seconde est strictement positive.

En prenant en compte cette condition et en traçant les dérivées (Figure 3.11b) d'une fonction gaussienne mono dimensionnelle $g(u)$ (Figure 3.11a), on remarque qu'elle n'est convexe que si $u < -\sigma$ et $u > \sigma$ (Figure 3.11c). Or, pour le cadrage il faut maximiser $g(u)$. On considère alors son l'opposé $-g(u)$ et la condition de convexité devient :

$$-\sigma < u < \sigma \quad (3.37)$$

En utilisant une image, chaque pixel a la forme suivante :

$$g_j(\mathbf{u}, \boldsymbol{\mu}_j, \sigma_j) = \exp \left(-\frac{\|\mathbf{u} - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2} \right) \quad (3.38)$$

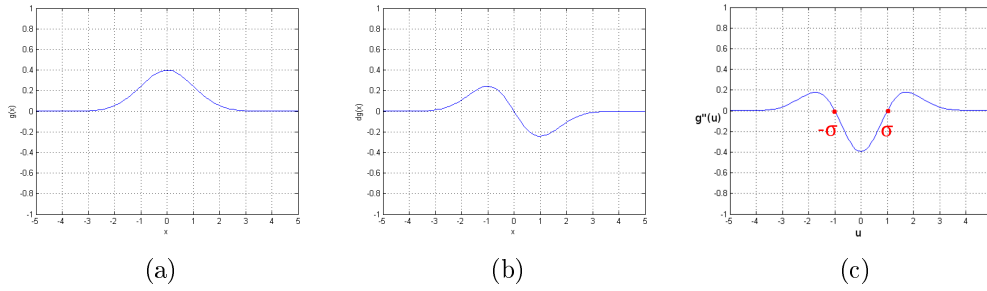


FIGURE 3.11 – (a) Fonction Gaussienne avec $\sigma = 1$, (b) première dérivée de la gaussienne, (c) seconde dérivée de la gaussienne.

La pondération de l'exponentielle n'a pas été prise en compte afin de simplifier l'expression de la gaussienne et de réduire les temps de calcul, en pratique.

Ainsi, pour l'opposé de la fonction gaussienne, la condition de convexité est donnée en combinant les équations (3.37) et (3.36) :

$$-\lambda I_s(\boldsymbol{\mu}_j) < \|\mathbf{u} - \boldsymbol{\mu}_j\| < \lambda I_s(\boldsymbol{\mu}_j) \quad (3.39)$$

Il est question de cadrer l'information de saillance dans l'image, donc $\lambda I_s(\boldsymbol{\mu}_j)$ doit être égal soit à la demi-largeur de l'image soit à sa demi-hauteur.

Si on choisit de réaliser le cadrage proportionnellement à la largeur W de l'image, on pose :

$$\lambda I_s(\boldsymbol{\mu}_j) = \frac{W}{2} \quad (3.40)$$

Ainsi, on obtient pour une gaussienne l'expression suivante de λ :

$$\lambda = \frac{W}{2I_s(\boldsymbol{\mu}_j)} \quad (3.41)$$

Cependant, l'image est modélisée par un mélange de gaussiennes. Pour calculer un seul λ pur toutes les gaussiennes, on remplace les différents $I_s(\boldsymbol{\mu}_j)$ par la moyenne des intensités de saillance dans l'image, \bar{I}_s :

$$\bar{I}_s = \frac{1}{N_j} \sum_j I_s(\boldsymbol{\mu}_j) \quad (3.42)$$

où N_j est le nombre de pixels saillants dans l'image.

Finalement, λ est donné par :

$$\lambda = \frac{W}{2\bar{I}_s} \quad (3.43)$$

Par exemple, en utilisant cette méthode pour le calcul de λ , et en considérant l'image représentée par la figure 3.9a, le λ résultant en utilisant l'équation (3.43) est 0.19. La figure 3.12 montre l'allure du mélange de gaussiennes obtenu.

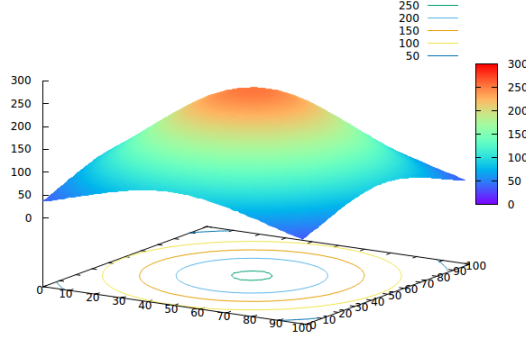


FIGURE 3.12 – Le mélange de gaussiennes associé à l'image en utilisant la méthode proposée pour le calcul du λ .

Toutes les expérimentations des sections 3.3 et 3.4 utilisent cette méthode de calcul de λ .

3.3.5 Résolution du problème

On commence par calculer la carte de saillance I_s de l'image rendue à la pose courante \mathbf{r}_k en utilisant l'approche [Radhakrishna 2009]. Ensuite, chaque pixel saillant est transformé en une gaussienne. La distribution de saillance dans l'image est modélisée par le mélange de gaussiennes basées saillance. La fonction de coût utilisée G (Équation (3.35)) est encore une fois non linéaire, la même démarche que celle présentée dans la section 3.2 est adoptée. Le but est de maximiser $G(\mathbf{r})$, ce qui est équivalent à la minimisation de son opposée $-G(\mathbf{r})$. On cherche alors à annuler la Jacobienne à la pose \mathbf{r}_k :

$$\frac{\partial(-G(\mathbf{r}_k))}{\partial \mathbf{r}} = 0 \quad (3.44)$$

On note par $d_{\mathbf{r}}(-G(\mathbf{r}_k))$ la dérivée partielle du premier ordre de la fonction $-G(\mathbf{r}_k)$ par rapport à \mathbf{r}_k et par $d_{\mathbf{r}}^2(-G(\mathbf{r}_k))$ la dérivée partielle seconde. Le développement de Taylor au premier ordre de l'équation (3.44) donne :

$$d_{\mathbf{r}}(-G(\mathbf{r}_k)) = d_{\mathbf{r}}(-G(\mathbf{r}_{k-1})) + d_{\mathbf{r}}^2(-G(\mathbf{r}_{k-1}))\dot{\mathbf{r}}_k \quad (3.45)$$

Pour résoudre cette équation, la Jacobienne et le Hessien de la fonction $-G(\mathbf{r}_{k-1})$ doivent être calculés. Un avantage majeur de l'utilisation du mélange de gaussiennes

réside dans la possibilité de pouvoir calculer analytiquement la solution exacte du gradient en dérivant l'équation 3.38 :

$$d_{\mathbf{r}}(-G(\mathbf{r}_{k-1})) = - \sum_{\mathbf{u}} \sum_j \frac{\partial g(\mathbf{u}, \boldsymbol{\mu}_j, \sigma_j)}{\partial \boldsymbol{\mu}_j} \mathbf{L}_{\boldsymbol{\mu}_j} \quad (3.46)$$

où $\mathbf{L}_{\boldsymbol{\mu}_j}$ est la matrice d'interaction qui lie la variation de la j -ième gaussienne dans l'image à la pose de la caméra. Cette matrice est la même que celle définie par l'équation (3.22).

Le gradient de la fonction gaussienne (Équation (3.38)) est donné par :

$$\frac{\partial g(\mathbf{u}, \boldsymbol{\mu}_j, \sigma_j)}{\partial u_j} = \frac{(u - u_j)}{\sigma_j^2} \exp^{-\frac{\|\mathbf{u} - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2}} \quad (3.47)$$

et,

$$\frac{\partial g(\mathbf{u}, \boldsymbol{\mu}_j, \sigma_j)}{\partial v_j} = \frac{(v - v_j)}{\sigma_j^2} \exp^{-\frac{\|\mathbf{u} - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2}} \quad (3.48)$$

Pour le calcul du Hessian, on obtient :

$$d_{\mathbf{r}}^2(-G(\mathbf{r}_{k-1})) = - \sum_{\mathbf{u}} \sum_j \mathbf{H}_{G_j} \quad (3.49)$$

Dans cette dernière équation :

$$\mathbf{H}_{G_j} = \mathbf{L}_{\boldsymbol{\mu}_j}^{\top} \frac{\partial^2 g(\mathbf{u}, \boldsymbol{\mu}_j, \sigma_j)}{\partial \boldsymbol{\mu}_j^2} \mathbf{L}_{\boldsymbol{\mu}_j} + \frac{\partial g(\mathbf{u}, \boldsymbol{\mu}_j, \sigma_j)}{\partial u_j} \mathbf{H}_{u_j} + \frac{\partial g(\mathbf{u}, \boldsymbol{\mu}_j, \sigma_j)}{\partial v_j} \mathbf{H}_{v_j} \quad (3.50)$$

où \mathbf{H}_{u_j} et \mathbf{H}_{v_j} sont les Hessiennes des deux coordonnées du pixel $\boldsymbol{\mu}_j$ définies par les matrices 6×6 des équations (3.27) et (3.28). La dérivée partielle du second ordre de la fonction gaussienne par rapport à $\boldsymbol{\mu}_j$ est donnée par :

$$\frac{\partial^2 g(\mathbf{u}, \boldsymbol{\mu}_j, \sigma_j)}{\partial \boldsymbol{\mu}_j^2} = \begin{pmatrix} \left(\frac{1}{\sigma_j^2} - \frac{(u-u_j)^2}{\sigma_j^4} \right) & -\frac{(u-u_j)(v-v_j)^2}{\sigma_j^4} \\ -\frac{(u-u_j)(v-v_j)}{\sigma_j^4} & \left(\frac{1}{\sigma_j^2} - \frac{(v-v_j)^2}{\sigma_j^4} \right) \end{pmatrix} \exp^{-\frac{\|\mathbf{u} - \boldsymbol{\mu}_j\|^2}{2\sigma_j^2}} \quad (3.51)$$

La loi de commande finale du mélange de gaussiennes basées saillance est donnée par :

$$\dot{\mathbf{r}}_k = \beta (d_{\mathbf{r}}^2(-G(\mathbf{r}_{k-1})))^+ d_{\mathbf{r}}(-G(\mathbf{r}_{k-1})) \quad (3.52)$$

où β est un paramètre scalaire représentant le gain, il permet d'ajuster l'amplitude du mouvement.

3.3.6 Résultat de simulation : environnement 2D

Afin de montrer l'avantage de l'utilisation de la nouvelle modélisation des mélanges de gaussiennes basées sur la saillance comme primitive visuelle, la même expérience illustrée précédemment pour la l'entropie photométrique dans un environnement 2D a été menée. A partir d'une pose initiale identique à celle utilisée dans la section 3.2 représentée avec le rectangle (cadre) en pointillé dans la figure 3.13, l'algorithme d'asservissement cherche à maximiser l'information de pertinence (mélange de gaussiennes basées saillance) dans l'image. Par ailleurs, on constate d'après la figure 3.13 que le cadrage final inclut toute l'information pertinente (la face globale dans cet exemple), on observe aussi que le mélange de gaussiennes est centré quand la convergence est atteinte (sous figure nommée "MG final").

Un deuxième exemple illustré par la figure 3.14 montre que même à partir d'une pose initiale très pauvre en information visuelle (un nombre très restreint de pixels saillants inclus dans le cadre initial), l'algorithme réussit à converger avec succès. En effet, l'information de pertinence qui est représentée par le visage dans cette image a rempli le cadrage final. La trajectoire de la caméra est optimale et quelques itérations seulement suffisent à bien cadrer le visage d'où la rapidité de la convergence.

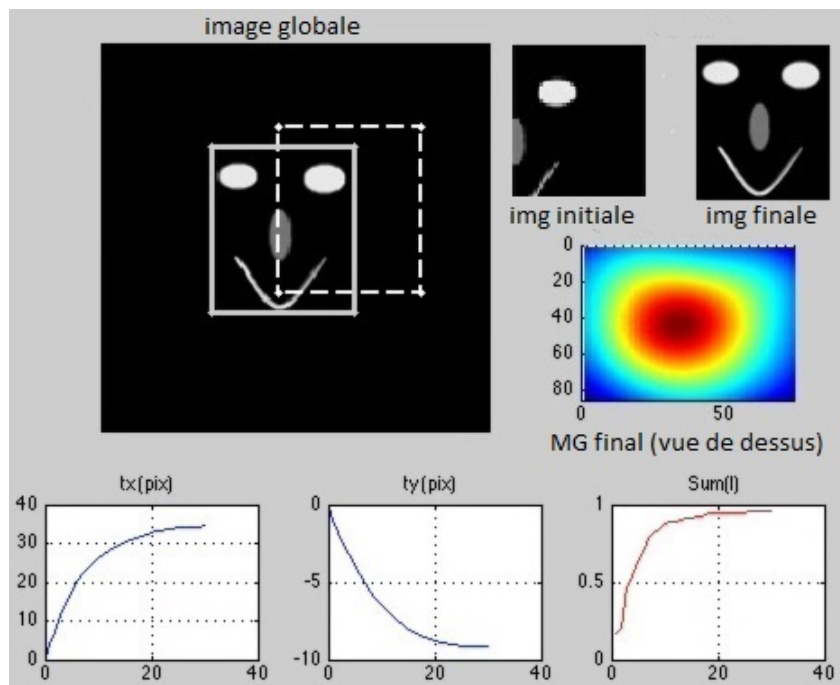


FIGURE 3.13 – Cadrage visuel : mélange de gaussiennes basées saillance.

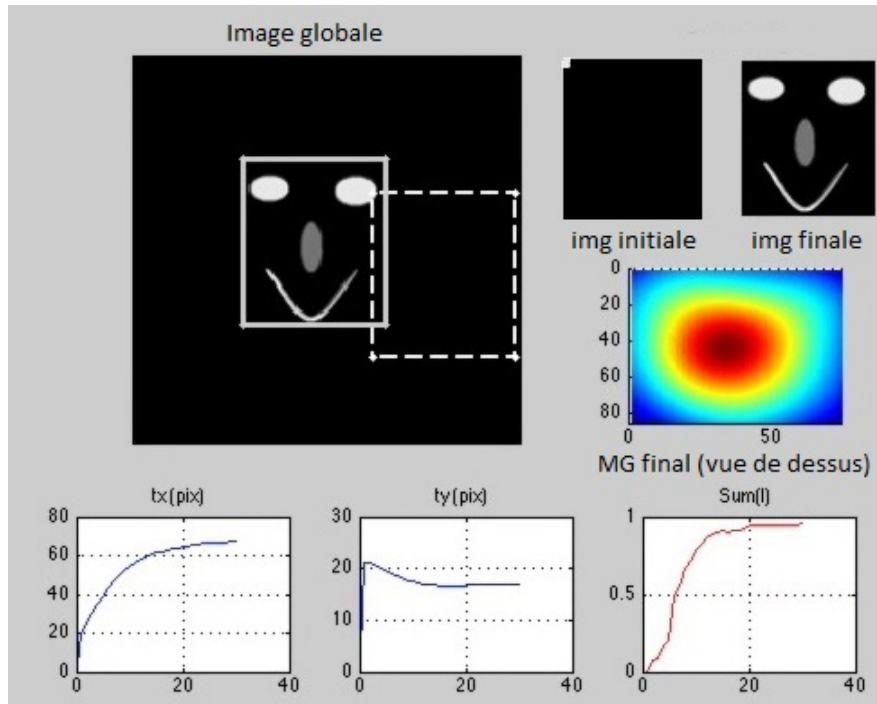
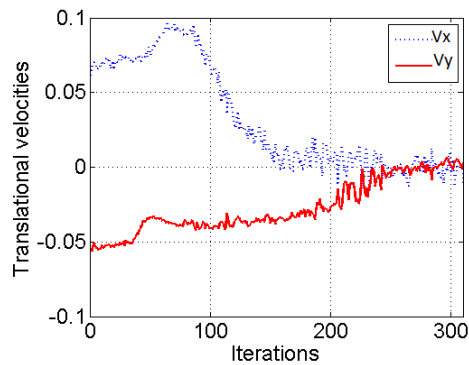
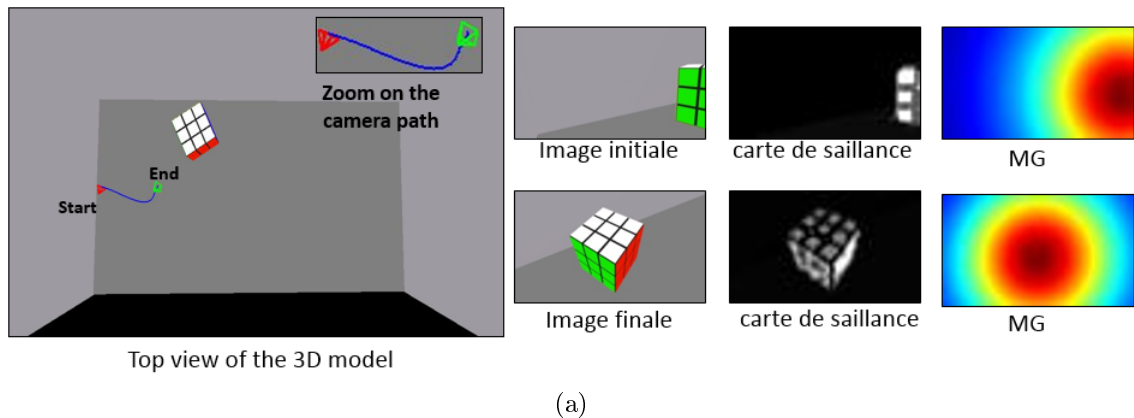


FIGURE 3.14 – Cas extrême de cadrage en utilisant la primitive mélange de gaussiennes basées saillance.

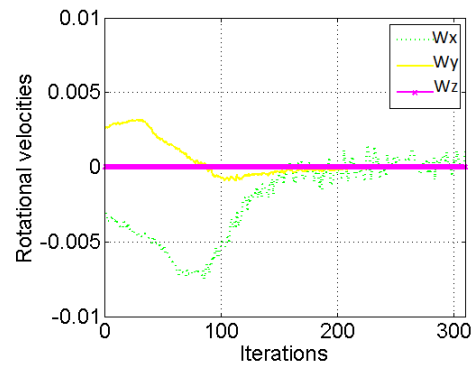
3.3.7 Résultat de simulation : environnement 3D

Dans un environnement 3D simple de synthèse composé d'une pièce avec un cube de Rubik, on introduit une navigation purement automatique en asservissant tous les d.d.l de la caméra mis part la translation t_z sur l'axe \vec{Z} de la caméra virtuelle. Cette translation a été bloquée car d'une part la maximisation du mélange de gaussiennes basées saillance va amener la caméra à s'approcher de l'objet qu'on veut cadrer. Dès lors, l'objet va remplir tout le champ de vu de la caméra (ce n'est pas toujours pertinent car cela dépend du niveau de détails de l'objet). D'autre part, on voudrait donner la liberté à l'utilisateur de s'approcher ou bien, de s'éloigner de l'objet. Ceci est expliqué dans la section 3.4. Une vue de dessus de l'environnement 3D accompagnée des images initiale et finale rendues, leur cartes de saillance et les mélanges correspondants sont illustrées dans la figure 3.15. Ainsi, cette expérience montre que l'algorithme converge vers un cadrage final très satisfaisant en utilisant uniquement les cinq autres d.d.l. En effet, le point de vue pertinent atteint par la caméra permet de visualiser trois faces du cube.

Les figures 3.15b et 3.15c montrent les courbes de vélocité, qui sont bruitées et



(b)



(c)

FIGURE 3.15 – loi de commande mélange de gaussiennes basées saillance dans un environnement 3D simple.

cela est dû à l'utilisation d'images de très basse résolution 60×30 pour accélérer le calcul. Cependant, ce bruit n'affecte ni la qualité du rendu ni la trajectoire de la caméra qui reste fluide et non saccadée.

3.3.7.1 Impact des d.d.l de rotations

Si on voulait juste centrer l'information de pertinence dans l'image, on pourrait juste faire un asservissement visuel en utilisant par exemple le barycentre de la zone de saillance comme primitive géométrique. Cette méthode va donc assurer de garder la région saillante centrée dans l'image, mais elle ne garantit pas de la visualiser sous le meilleur angle de vue. Il est à rappeler que d'après notre définition du cadrage (Définition 1), il ne suffit pas juste de centrer l'information mais aussi d'assurer une bonne orientation de la caméra. C'est la raison pour laquelle il faut aussi agir sur les rotations.

Pour illustrer l'impact de l'utilisation des d.d.l de rotations, un environnement 3D simple constitué d'un fauteuil est considéré et deux expériences sont menées. Dans la première, le cadrage visuel est réalisé à l'aide d'un asservissement visuel ne portant que sur les deux translations de la caméra suivant les axes \vec{X} et \vec{Y} . Comme illustré par la figure 3.16, le fauteuil est centré dans l'image mais la caméra a gardé son orientation initiale qui n'a pas permis de bien visualiser l'objet une fois centré. Par contre, quand le cadrage est accompli en asservissant sur tous les d.d.l de la caméra mis à part la translation sur l'axe \vec{Z} , il est clair (figure 3.17) que les rotations sont corrigées ce qui améliore la précision du cadrage. La valeur de la fonction de coût une fois la convergence atteinte est plus élevée quand les rotations sont contrôlées en comparant entre les figures 3.16b et 3.17b. Par conséquent, l'utilisation des mélanges de gaussiennes basées saillance en utilisant les cinq d.d.l de la caméra fournit un point de vue "pertinent" pour la visualisation du fauteuil. D'autres expérimentations ont confirmé cet avantage, ce qui rend cette primitive adéquate avec le problème du cadrage visuel.

3.3.7.2 Impact de la résolution des images

La loi de commande calculée à partir de l'optimisation des mélanges de gaussiennes basées saillance tient compte du nombre de pixels saillants dans l'image. Ainsi, la taille des images utilisées a une influence sur le calcul. Or, le calcul du mélange de gaussiennes augmente le temps de calcul. La résolution des images utilisées pour le calcul est baissée car elle n'influe pas sur le comportement de la caméra et n'a pas d'impact sur la convergence.

Pour expliquer ce comportement, une image rendue à partir d'un nuage de points 3D de synthèse à différentes résolutions (Figure 3.18a) est considérée. Pour chacune de ces images, l'allure du mélange de gaussiennes en 3D (Figure 3.18) est tracée. D'après la Figure 3.18, on remarque que le mélange de gaussiennes calculé pour la même image à différentes résolutions est quasiment similaire. Malgré la taille de l'image qui change, la surface qu'occupe le mélange de gaussiennes basées saillance est presque la même proportionnellement à la résolution de l'image. Ce qui prouve que l'utilisation de l'image qui a pour mélange de gaussiennes la figure 3.18b ou celle qui a la figure 3.18f aboutit à la même loi de commande. L'utilisation du λ proportionnel à la largeur de l'image permet d'avoir un mélange de gaussiennes avec un seul mode principal pour chaque image et non un mélange avec plusieurs modes. Ainsi, l'image est représentée avec le même mélange quelque soit sa résolution.

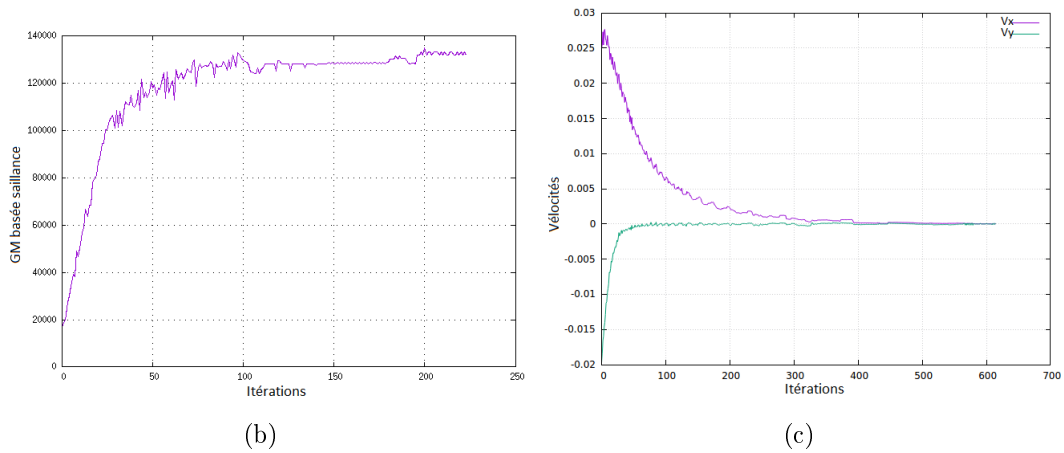
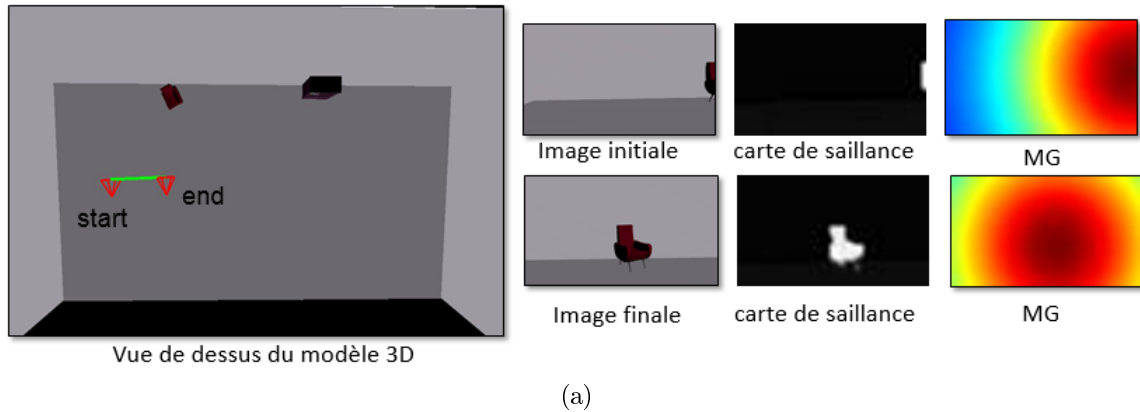


FIGURE 3.16 – Navigation automatique sans rotation en utilisant la primitive mélange de gaussiennes basées saillance.

3.3.8 Résultat sur un nuage de points dense et coloré

Dans un environnement à nuage de points 3D, le but est de garantir un mouvement réaliste en plus de la pertinence visuelle. Pour cela, une contrainte d'évitement d'obstacles ainsi que la suppression de rotations irréalistes autour de l'axe optique de la caméra (le "Roulis") ont été intégrées.

3.3.8.1 Évitement d'obstacles

Pour respecter cette contrainte, tous les points qui sont à une distance fixe par rapport à la caméra, par exemple $1m$ ou $2m$ ou plus seront pris en compte. Ensuite chaque point est considéré comme un obstacle à lui seul et le but est de maximiser la distance entre la caméra et tous les points. Pour ce faire, la fonction de coût introduite par [Courty 2001] qui tend vers l'infini quand la distance entre la caméra

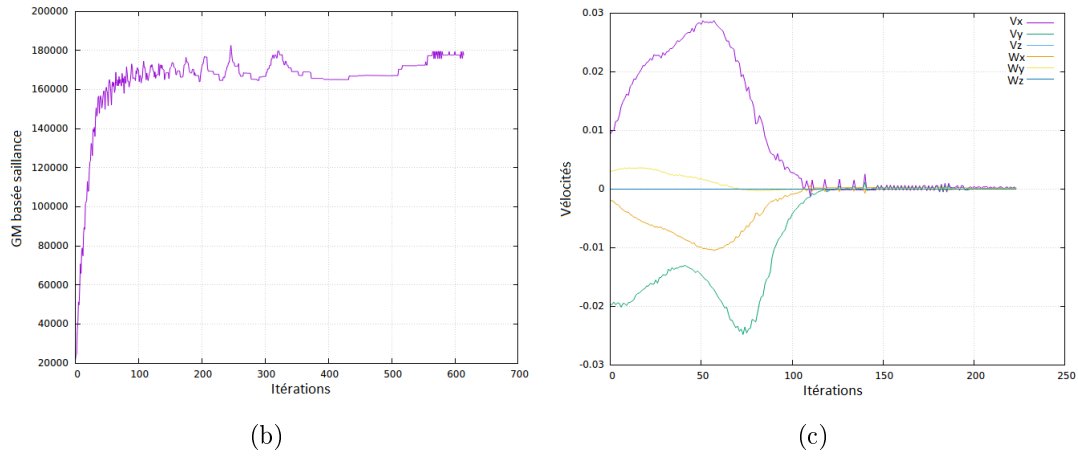
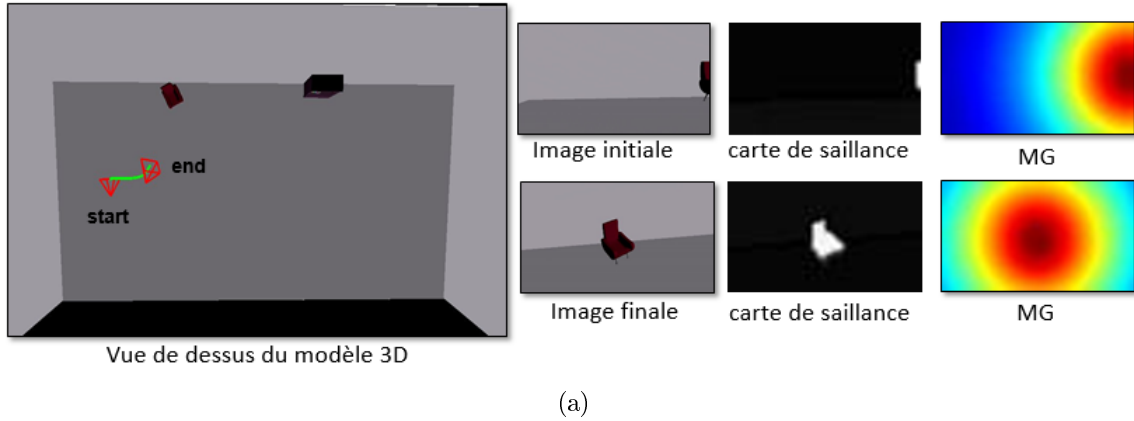


FIGURE 3.17 – Navigation automatique avec rotation en utilisant la primitive mélange de gaussiennes basées saillance.

et l'objet est nulle va être utilisée :

$$O(\mathbf{r}) = \sum_k \frac{1}{2\|{}^w\mathbf{p}_c - {}^w\mathbf{p}_{o_k}\|^2} \quad (3.53)$$

où k est l'index d'un point. ${}^w\mathbf{p}_c = (X_c, Y_c, Z_c)^T$ est la position de la caméra et ${}^w\mathbf{p}_{o_k} = (X_{o_k}, Y_{o_k}, Z_{o_k})^T$ la position d'un point. Ces deux positions sont exprimées dans le repère monde \mathfrak{R}_w .

Le but est d'éviter à la caméra de percuter un obstacle. Seule la position de la caméra (et non son orientation) sera prise en compte dans la tâche de positionnement parce qu'on suppose que la camera virtuelle est représentée par un point, notamment son centre optique. Pour considérer tous les points à une distance fixe de la caméra, c'est-à-dire ceux qui se trouvent dans une sphère ayant pour rayon la distance au centre la caméra, on a utilisé une caméra virtuelle équirectangulaire implémentée à l'aide d'un shader GPU.

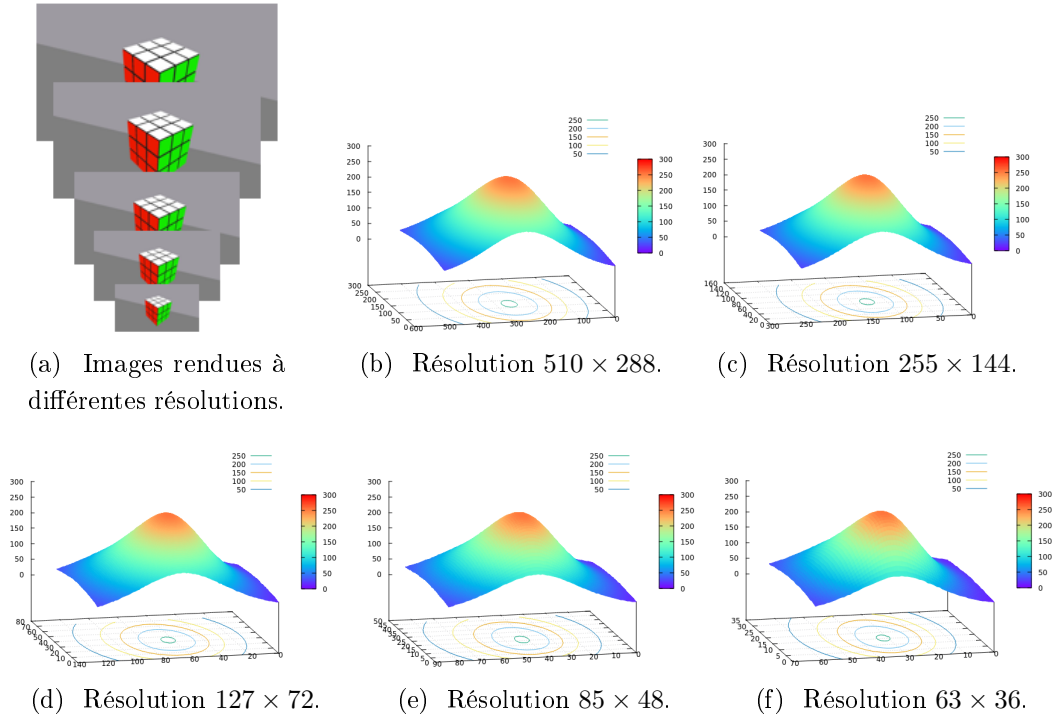


FIGURE 3.18 – Impact de la résolution de l'image sur le mélange de gaussiennes.

Pour obtenir une image équirectangulaire à partir d'une scène 3D, une projection sphérique est utilisée suivie d'une projection cylindrique. En effet, les points 3D de l'objet sont initialement projetés sur une sphère, ensuite l'azimut et l'élévation des points sur la sphère sont projetés sur un cylindre qui est déplié sous la forme d'un plan image.

La projection sphérique permet de transformer un point ${}^c\mathbf{X}$ en un point ${}^s\mathbf{X}$ sur la sphère unitaire :

$$\begin{bmatrix} {}^sX \\ {}^sY \\ {}^sZ \end{bmatrix} = \begin{bmatrix} {}^cX/\rho \\ {}^cY/\rho \\ {}^cZ/\rho \end{bmatrix} \quad (3.54)$$

où $\rho = \sqrt{({}^cX^2 + {}^cY^2 + {}^cZ^2)}$ est la distance entre le point ${}^c\mathbf{X}$ et le centre de la sphère. Ensuite une projection cylindrique est effectuée pour obtenir l'image équirectangulaire (Figure 3.19).

Étant donné un point sur la sphère ${}^s\mathbf{X}$, son azimut λ et son élévation ϕ sont calculés sur le plan de l'image polaire à l'aide des relations suivantes :

$$\begin{cases} \lambda = \text{Atan} \left(\frac{{}^sZ}{{}^sX} \right) \\ \phi = -2\text{acos} ({}^sY) \end{cases} \quad (3.55)$$

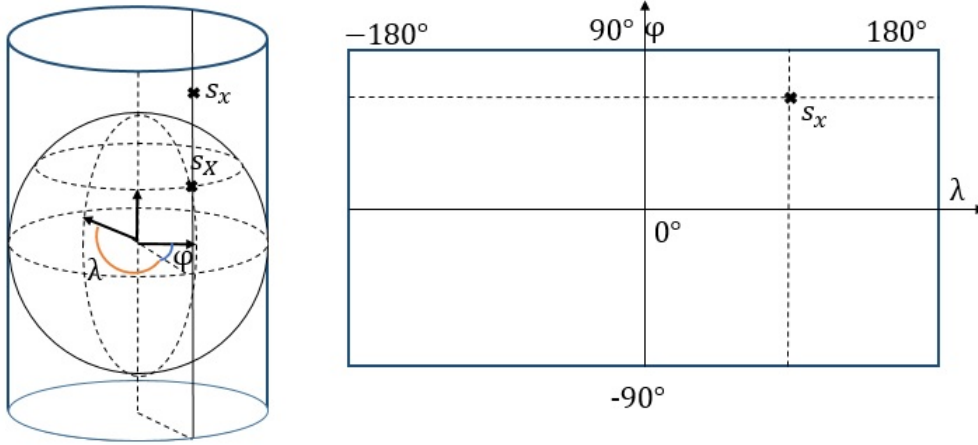


FIGURE 3.19 – Projection cylindrique.

Finalement, les coordonnées pixeliques \mathbf{u} sont obtenues après application de la matrice des paramètres intrinsèques \mathbf{K} .

A chaque itération de l'algorithme, la caméra equirectangulaire virtuelle permet d'obtenir une image, ce qui permet de récupérer les coordonnées des points des obstacles qui se trouvent à une distance fixe (qu'on a choisi) de la caméra. Ils seront utilisés par la suite pour le calcul de la loi de commande basée évitement d'obstacles. Pour la résolution, l'algorithme de descente du gradient est exploité. Le développement de Taylor est utilisé pour la linéarisation de la fonction :

$$O(\mathbf{r}_k) \simeq O(\mathbf{r}_{k-1}) + d_{\mathbf{r}}(O(\mathbf{r}_{k-1}))\dot{\mathbf{r}}_k \quad (3.56)$$

La maximisation du mélange de gaussiennes basées saillance ainsi que la maximisation de la distance entre la caméra et les obstacles sont fusionnées dans une loi de commande hybride [Hager 1997] en empilant le Hessien et la Jacobienne des deux lois de contrôle. Ceci permet d'éviter les changements soudains et répétés de la direction du mouvement lors de la détection d'un obstacle. L'estimation de l'incrément de pose hybride $\dot{\mathbf{r}}_k$ est donnée par :

$$\dot{\mathbf{r}}_k = \begin{pmatrix} d_{\mathbf{r}}^2(-G(\mathbf{r}_{k-1})) \\ d_{\mathbf{r}}(O(\mathbf{r}_{k-1})) \end{pmatrix}^+ \begin{pmatrix} d_{\mathbf{r}}(-G(\mathbf{r}_{k-1})) \\ -O(\mathbf{r}_{k-1}) \end{pmatrix} \quad (3.57)$$

3.3.8.2 Gestion des rotations

La navigation automatique dans un environnement 3D peut générer des mouvements irréalistes autour de l'axe optique de la caméra et plus précisément le "Roulis". Dès lors, la caméra n'aura que cinq d.d.l opérationnels $\mathbf{q}_k = (t_X, t_Y, t_Z, \theta_p, \theta_t)$,

tel que (t_X, t_Y, t_Z) sont les trois translations et (θ_p, θ_t) les deux rotations correspondantes au mouvement du "panorama" et de "l'inclinaison" au lieu des six d.d.l cartésiens généralement considérés en asservissement visuel.

Pour cela, on assimile la caméra à un robot virtuel avec seulement cinq d.d.l, où les trois translations sont réalisées dans le repère cartésien (ou le repère monde \mathfrak{R}_w), alors que les deux rotations sont réalisées successivement dans le repère caméra \mathfrak{R}_c . On note par \mathbf{q}_k la position du robot virtuel à l'itération k .

Le Jacobien du robot virtuel ${}^c\mathbf{J}_c$ dans le repère \mathfrak{R}_c est exprimé sous la forme :

$${}^c\mathbf{J}_c = {}^c\mathbf{V}_w {}^w\mathbf{J}_w \quad (3.58)$$

où ${}^w\mathbf{J}_w$ est le Jacobien du robot dans le repère \mathfrak{R}_w et ${}^c\mathbf{V}_w$ une matrice de changement de repère cinématique (de \mathfrak{R}_w à \mathfrak{R}_c).

Ainsi, l'équation (3.57) devient :

$$\dot{\mathbf{q}}_k = \left(\begin{pmatrix} d_{\mathbf{r}}^2(-G(\mathbf{r}_{k-1})) \\ d_{\mathbf{r}}(O(\mathbf{r}_{k-1})) \end{pmatrix} {}^c\mathbf{J}_c \right)^+ \begin{pmatrix} d_{\mathbf{r}}(-G(\mathbf{r}_{k-1})) \\ -O(\mathbf{r}_{k-1}) \end{pmatrix} \quad (3.59)$$

tel que $\dot{\mathbf{q}}_k$ est l'incrément de pose des cinq d.d.l opérationnels. Sa dimension est cinq car pour le calculer on multiplie le Hessien des deux contraintes $\begin{pmatrix} d_{\mathbf{r}}^2(-G(\mathbf{r}_{k-1})) \\ d_{\mathbf{r}}(O(\mathbf{r}_{k-1})) \end{pmatrix}$ de taille $(N \times 6)$, tel que N est le nombre de pixels saillants et de points, par le jacobien qui est de taille (6×5) .

La matrice de transformation est construite à partir du vecteur de translation ${}^c\mathbf{t}_w$ et de la matrice de rotation ${}^c\mathbf{R}_w$, avec ${}^c\mathbf{R}_w = R_X(\theta_t)R_Z(\theta_p)$ où ${}^c\mathbf{R}_w = [\mathbf{q}_k[3], \mathbf{q}_k[4]]$ et le vecteur de translation ${}^c\mathbf{t}_w = -{}^c\mathbf{R}_w {}^c\mathbf{t}_w$ tel que ${}^c\mathbf{t}_w = (\mathbf{q}_k[0], \mathbf{q}_k[1], \mathbf{q}_k[2])^T$.

\mathbf{q}_k est calculé à l'aide de la relation :

$$\mathbf{q}_k = \mathbf{q}_{k-1} + \dot{\mathbf{q}}_k \quad (3.60)$$

3.3.8.3 Résultats sur le portail sud de la cathédrale d'Amiens

Cette approche a été testée sur un modèle 3D composé d'un nuage de points colorés : le portail sud de la cathédrale d'Amiens (Figure. 3.20). Le modèle a été numérisé à l'aide du scanner laser Faro. L'environnement n'a pas été numérisé dans sa totalité. Les parties blanches dans l'image rendue correspondent à des zones non scannées (des occultations, ...). Le moteur graphique Ogre 3D [Ogr] a été utilisé pour la visualisation du portail.

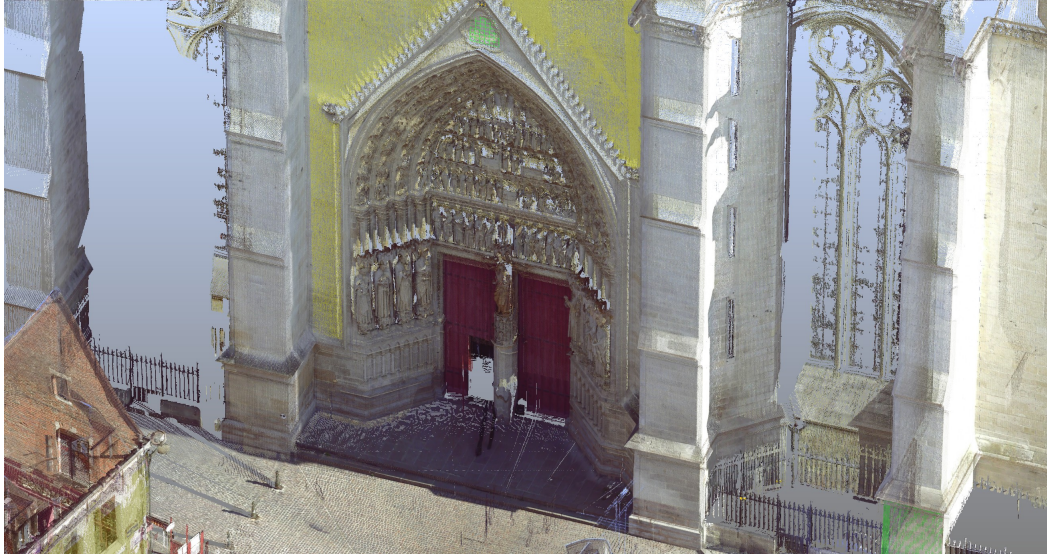


FIGURE 3.20 – L’environnement expérimental 3D du portail sud de la cathédrale.

Dans la figure 3.21a, on observe une vue de l’environnement 3D sur la gauche ainsi que les images initiale et finale rendues accompagnées de leurs cartes de saillance et leurs cartes de mélange de gaussiennes basées saillance correspondantes sur la gauche. Les vitesses de translation de la caméra sont représentées par les trois courbes de la figure 3.21b alors que celles des deux rotations sont représentées par la figure 3.21c.

La figure 3.22 montre que l’algorithme permet à la caméra de converger avec un cadrage visuel pertinent malgré le fait, que la pose initiale soit très pauvre en informations visuelles.

Ces résultats ont été obtenus à l’aide de la loi de commande hybride de l’équation (3.59) décrite précédemment. On remarque qu’on reste en accord avec les résultats de simulation, dans la mesure où l’algorithme converge vers un point de vue plus pertinent que l’image initiale.

3.3.9 Résultat en environnement réel

Afin de valider l’approche proposée dans un environnement réel, nous avons réalisé quelques expérimentations en utilisant un bras robot Staubli (Figure 3.23) à 6 d.d.l muni d’une caméra perspective montée sur son effecteur. Le développement a été fait en C++ avec les bibliothèques OpenCV et Visp [Marchand 2005]. Le but de ces expériences est de valider le cadrage visuel automatique en utilisant la représenta-

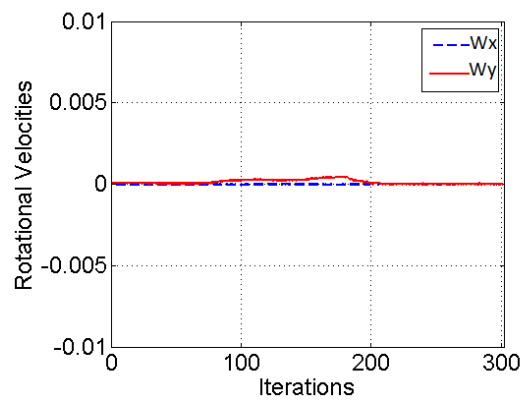
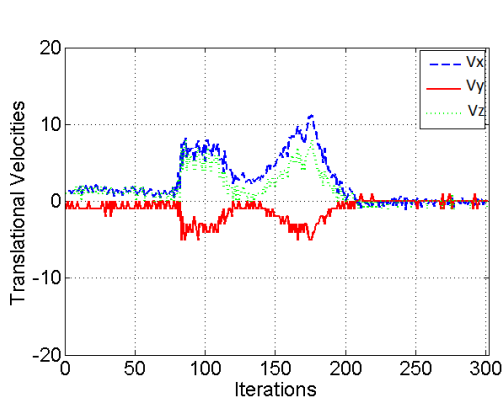
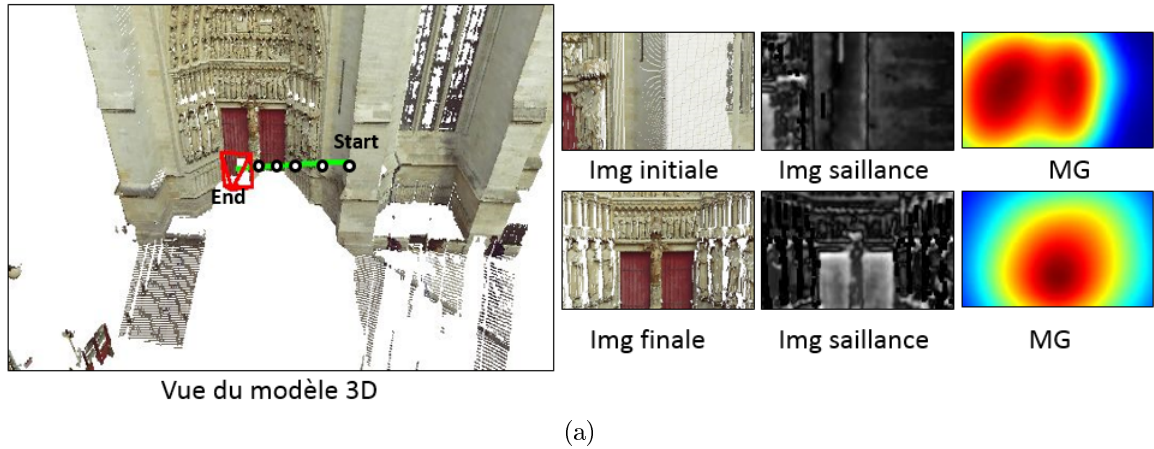
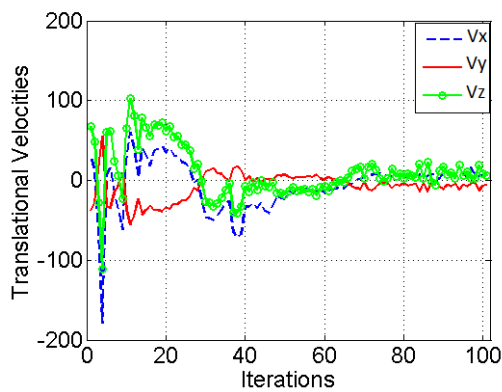
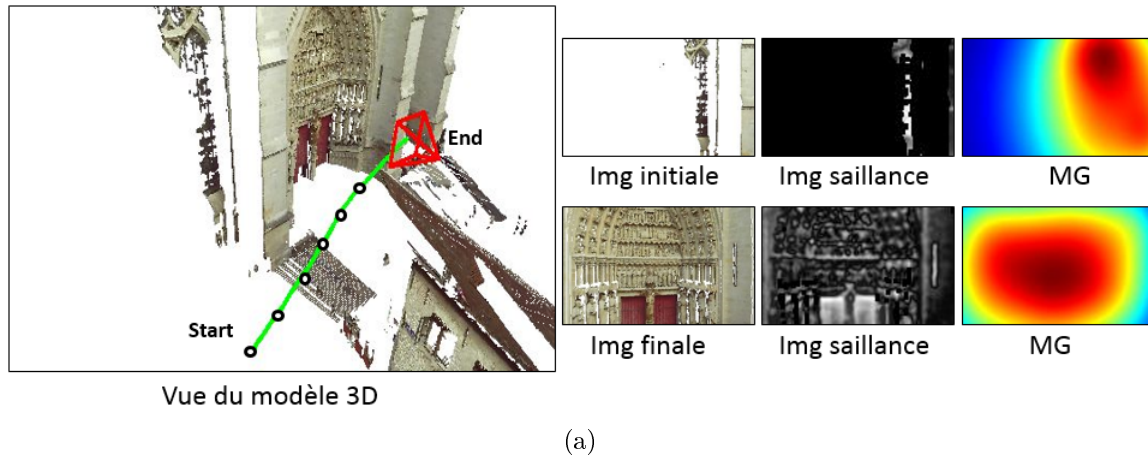


FIGURE 3.21 – Loi de commande hybride.

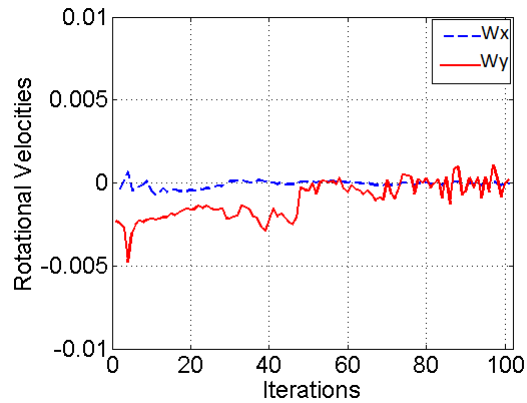
tion faite dans la section 3.3.

Seuls cinq d.d.l (la translation sur \vec{Z} a été bloquée) ont été contrôlés afin d'éviter que le robot ne s'approche trop de l'objet considéré pour le cadrage. Parmi les avantages de la méthode proposée, la possibilité de réduire la résolution des images permet de réduire les temps de calculs sans pour autant détériorer la qualité des résultats comme on a pu voir dans la section précédente.

Dans les expériences qui vont suivre, la taille des images utilisées pour le calcul des mélanges de gaussiennes est fixée à 64×51 pixels. Pour le déroulement de l'expérience, le robot démarre d'une position initiale \mathbf{r}_k et acquiert l'image correspondante $I(\mathbf{r}_k)$. Ensuite la loi de commande calculée à partir de l'équation (3.52) est envoyée au robot pour générer un déplacement et ainsi une nouvelle image est acquise. Ce processus est exécuté d'une manière itérative jusqu'à ce que la convergence soit atteinte. On remarque que le mélange de gaussiennes est centré dans la



(b)



(c)

FIGURE 3.22 – loi de commande hybride.

carte de le mélange de gaussiennes basées saillance une fois la convergence atteinte (Figure 3.24 et Figure 3.26), ce qui valide expérimentalement le cadrage visuel.

Durant cette expérience, à chaque fois que le robot cadre l'objet considéré (la convergence est atteinte vers le meilleur point de vue, celui qui inclut tout l'objet dans l'image et le centre), on déplace ce dernier manuellement afin d'étudier le comportement du robot après ce changement. En effet, l'algorithme prend en compte la nouvelle position de l'objet dans l'image et il tente encore une fois de le cadrer comme ceci est visible sur la séquence d'images de la figure 3.25. Ainsi l'algorithme est robuste aux changements brusques dans l'environnement.

Durant la tâche de navigation du robot, les conditions d'éclairage étaient variables. En effet, la lumière de la salle où se trouve le robot était soit éteinte soit allumée. Malgré cela, la tâche de cadrage était réalisée avec succès, même si ce résultat est davantage à mettre sur le compte de la méthode de calcul de la saillance



FIGURE 3.23 – Le robot utilisé dans nos expériences.

elle même. Ainsi, le critère des mélanges de gaussiennes basées saillance est robuste aux changements d'illumination. Finalement, en dépit du calcul de la matrice d'interaction en utilisant une profondeur constante, car on n'a pas l'information 3D, de l'environnement le cadrage visuel est accompli. Ainsi, cette approche est robuste par rapport à l'approximation de la profondeur. Les Figures 3.24 et 3.26 montrent le comportement du robot pour la tâche de cadrage visuel d'un objet plan et d'un autre non-plan.

3.4 Cadrage visuel assisté

Dans les sections précédentes de ce chapitre, la question étudiée concernait le contrôle automatique des d.d.l de la caméra. Nous allons maintenant proposer une méthode qui permet d'introduire l'utilisateur dans la navigation. La tâche principale qui consiste à cadrer l'information pertinente, repose sur la primitive mélange de gaussiennes basées saillance accompagnée d'une contrainte sur la distance (celle qui se trouve entre la caméra virtuelle et l'objet à cadrer) en plus de l'interaction de l'utilisateur. Cette interaction agira principalement sur les déplacements effectués le long de l'axe \vec{Z} de la caméra. Ainsi, en utilisant le clavier ou tout autre interface, l'utilisateur décide de sa proximité avec l'objet. Cependant, quand l'utilisateur n'effectue plus aucune interaction, la contrainte sur la distance est prise en compte pour contrôler automatiquement la proximité de la caméra par rapport à l'objet.

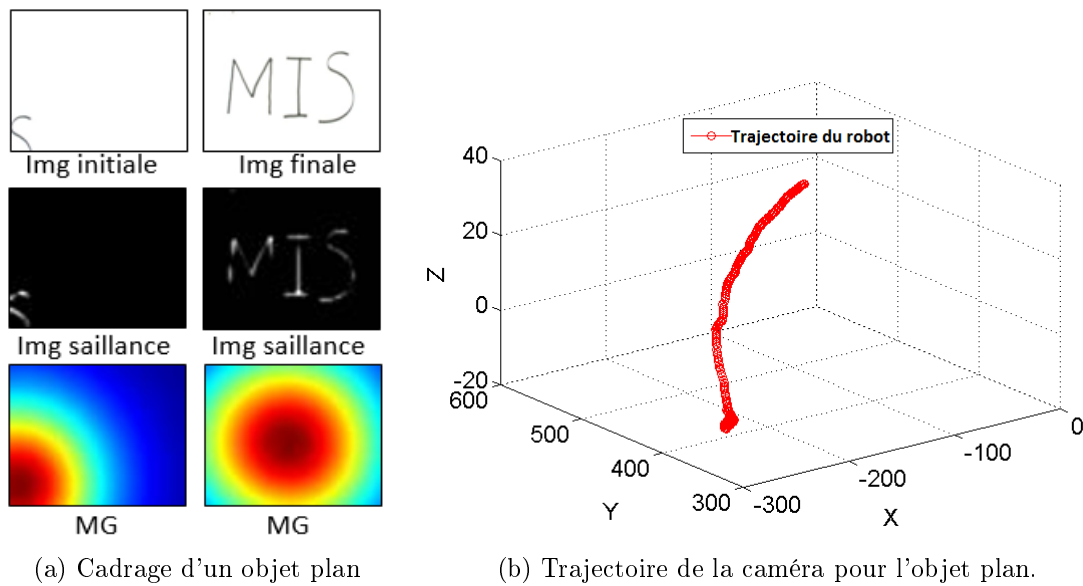


FIGURE 3.24 – Navigation automatique du robot.

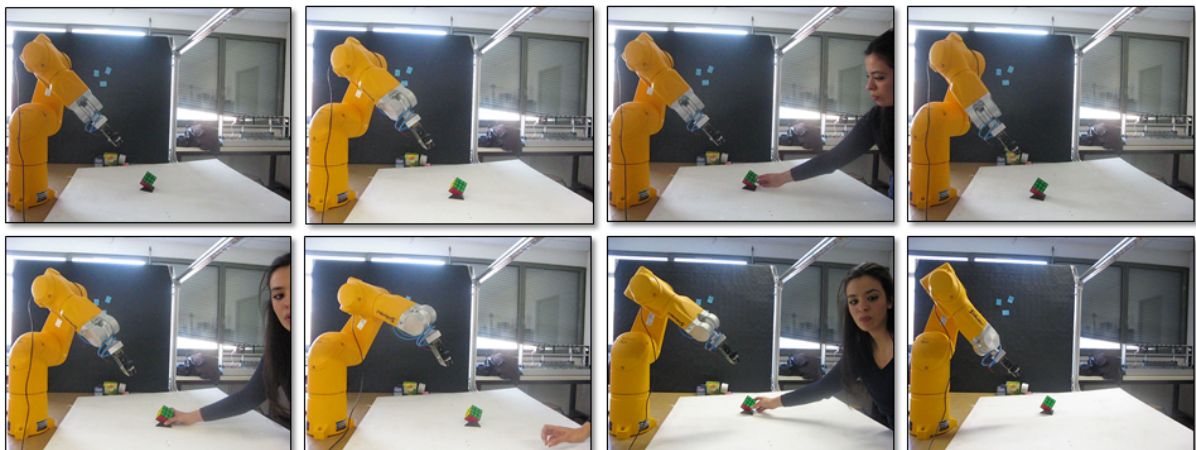


FIGURE 3.25 – Différentes positions atteintes par le robot durant le cadrage visuel.

3.4.1 Loi de commande assistée

On commence d'abord par combiner la maximisation des mélanges de gaussiennes basées saillance et la loi de commande basée distances sous la forme d'une loi hybride. La maximisation de la primitive visuelle mélange de gaussiennes basées saillance est effectuée de la même manière que la section précédente. L'incrément de pose (Équation (3.52)) calculé à partir de la maximisation de la fonction (Équa-

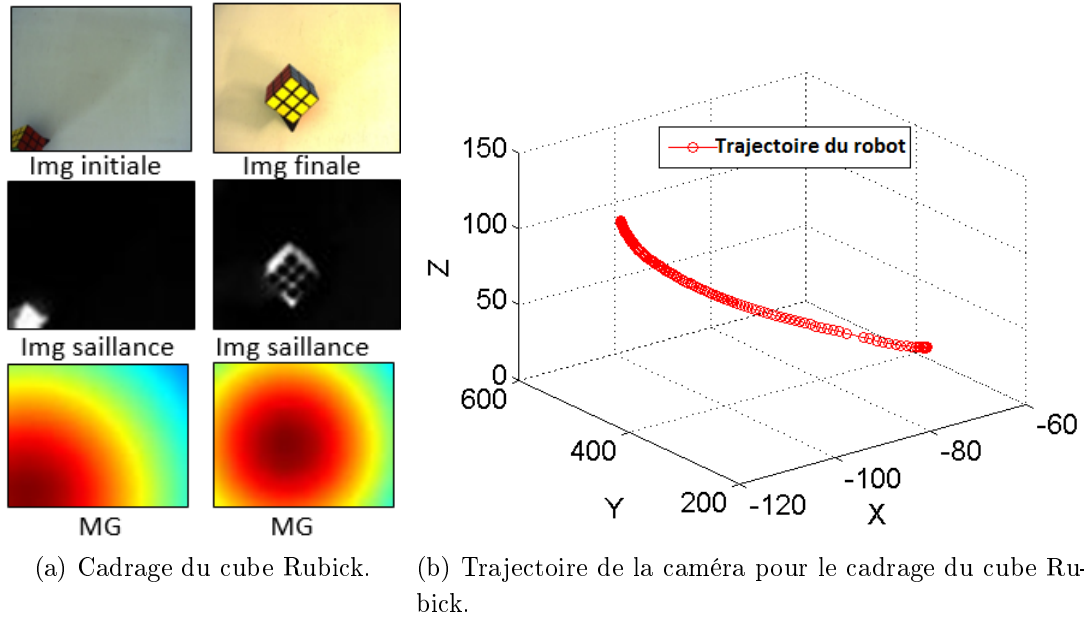


FIGURE 3.26 – Navigation automatique du robot.

tion (3.35)) va être utilisé. La loi basée distances représente la maximisation de la distance entre la caméra et l'objet à cadrer. Pour ce faire, on utilise la même fonction O (Equation (3.53)) définie pour l'évitement des obstacles. Cette fois ci, seule la translation selon l'axe \vec{Z} va être utilisée pour le calcul de la loi de commande basée distance :

$${}^D\dot{\mathbf{r}}_k = (d_{\mathbf{r}}O(\mathbf{r}_{k-1}))^+O(\mathbf{r}_{k-1}) \quad (3.61)$$

Ces deux lois de contrôle précédentes sont combinées afin d'obtenir l'incrément de pose hybride $\dot{\mathbf{r}}_k$:

$$\dot{\mathbf{r}}_k = {}^G\dot{\mathbf{r}}_k + \beta {}^D\dot{\mathbf{r}}_k \quad (3.62)$$

où ${}^G\dot{\mathbf{r}}_k$ est obtenu par l'équation (3.52) et $\beta > 0$ est une constante qui permet de combiner les deux lois de contrôle. La constante β est calculée à partir du ratio entre les normes de ${}^G\dot{\mathbf{r}}_k$ et ${}^D\dot{\mathbf{r}}_k$.

On a considéré une combinaison linéaire en pondérant les deux incréments. Ce choix a été fait empiriquement et validé expérimentalement. En plus de cette loi de commande hybride, une certaine liberté va être donnée à l'utilisateur durant la navigation dans un processus en ligne, qui consiste à modifier interactivement la distance séparant la caméra virtuelle et l'objet à cadrer. Ainsi, en utilisant une interface (clavier, souris, ...), l'utilisateur peut régler cette distance.

On note par ${}^U\dot{\mathbf{r}}_k$ l'incrément de pose généré par l'utilisateur (toutes ses composantes sont nulles mis-à-part l'incrément de la translation selon l'axe \vec{Z}). Cet

incrément sera combiné à la loi de commande définie $\dot{\mathbf{r}}_k$ (Équation (3.62)) afin d'obtenir ${}^A\dot{\mathbf{r}}_k$ (la loi de commande assistée) :

$${}^A\dot{\mathbf{r}}_k = \dot{\mathbf{r}}_k + \gamma^U \dot{\mathbf{r}}_k \quad (3.63)$$

où $\gamma > 0$ est une constante à déterminer empiriquement. Elle permet de modifier l'amplitude du mouvement et ainsi de régler la vitesse de convergence.

3.4.2 Résultat de simulation

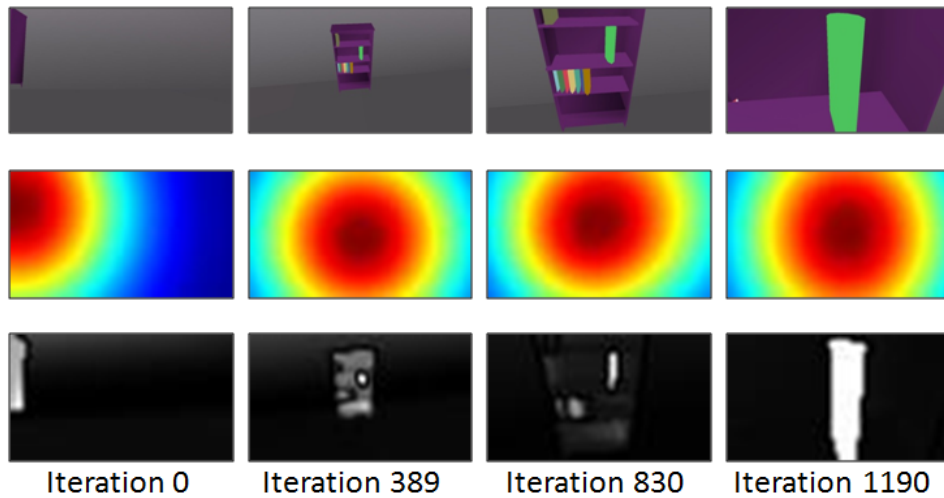
Pour évaluer l'approche de cadrage visuel assisté, une bibliothèque (Figure 3.27) constituée de plusieurs livres située dans une pièce est considérée comme environnement de test. La démarche consiste à cadrer la bibliothèque en se basant sur la maximisation des mélanges de gaussiennes basées saillance ainsi que la maximisation de la distance tout en prenant en compte le choix de l'utilisateur qui souhaiterait zoomer sur les livres, par exemple

La figure 3.27a montre quatre images clés de la navigation assistée. Dans cet exemple, on remarque que quand l'utilisateur s'approche de la bibliothèque, l'algorithme cherche à se focaliser sur le livre vert qui est le plus saillant d'après la carte de saillance. La figure 3.27b présente l'impact de l'utilisateur sur le changement de la vitesse de translation sur l'axe \vec{Z} . Ainsi, cette méthode simple montre l'intérêt de notre approche qui consiste à assister l'utilisateur en prenant en charge le contrôle des cinq d.d.l, tout en lui permettant de visualiser de près les détails qui l'intéressent.

3.4.3 Résultat sur un nuage de points 3D

L'approche a été aussi appliquée sur un modèle composé de nuages de points 3D colorés : une œuvre d'art constituée d'un tableau et d'une statue de la chapelle Saint Sébastien de la cathédrale Notre Dame d'Amiens. Elle a été numérisée avec précision en utilisant le scanner laser Leica C-10 et colorée à l'aide de la méthode de colorisation proposée par [Crombez 2014]. Pour le processus de rendu permettant de générer des vues synthétiques, le moteur graphique Ogre 3D [Ogr] a été utilisé. Cette partie illustre quelques résultats sur ce type de données.

Ayant choisi comme pose initiale de la caméra virtuelle, une posture très pauvre en information visuelle, l'algorithme de cadrage assisté est appliqué en asservissant tous les d.d.l de la caméra sauf la rotation autour de l'axe optique de la caméra \vec{Z} . Cette rotation a été bloquée afin d'éviter des mouvements irréalistes. Le cadrage



(a) Séquence d'images clés.

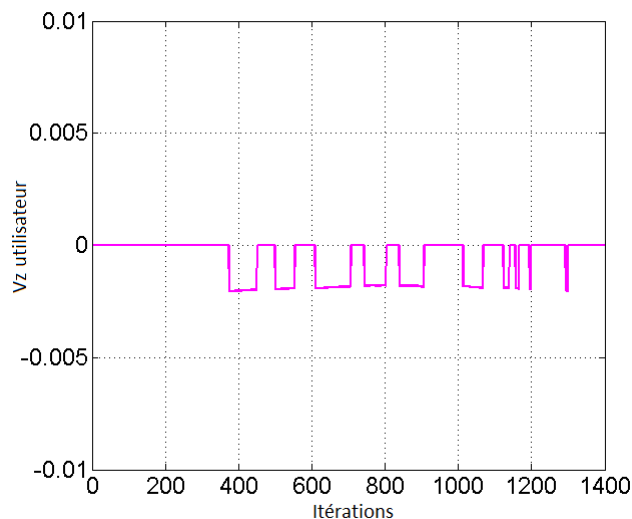
(b) La vitesse de translation sur l'axe \overrightarrow{z} .

FIGURE 3.27 – Cadrage visuel assisté dans un environnement de synthèse.

visuel est réalisé en utilisant la loi de commande définie par l'équation (3.63). En outre, il s'avère essentiel d'ajouter la contrainte qui borne la distance entre la caméra et l'objet ainsi que l'interaction de l'utilisateur. En effet, la loi de commande des mélanges de gaussiennes basées saillance a été conçue pour maximiser la pertinence dans l'image. La caméra risque donc de s'approcher de l'objet pour remplir toute l'image rendue. Ceci n'est pas toujours pertinent comme comportement.

Conformément à l'évaluation réalisée à la fin de cette section, la majorité des utilisateurs sont d'accord sur le fait qu'un "bon cadrage visuel" est un cadrage qui

permet de centrer l'information saillante (l'objet) dans l'image rendue. C'est pour cela que ce processus est automatisé. A contrario, le déplacement en direction des niveaux de détails dans une scène est une action très subjective, propre à chaque utilisateur, et par conséquent difficile à automatiser.

La figure 3.28 montre qu'à partir d'une pose initiale où la face de l'œuvre d'art est cachée (première image de la séquence), la caméra se déplace dans l'environnement afin d'arriver à se positionner parfaitement devant l'œuvre d'art en incluant dans son champ de vue toute l'information "pertinente", notamment le tableau et la statue. Par ailleurs, le mode principal du mélange de gaussiennes correspondant à l'image initiale est positionné à la partie gauche de la carte et légèrement décalé vers le haut, alors qu'il est parfaitement centré une fois la convergence atteinte (la dernière image de la séquence).

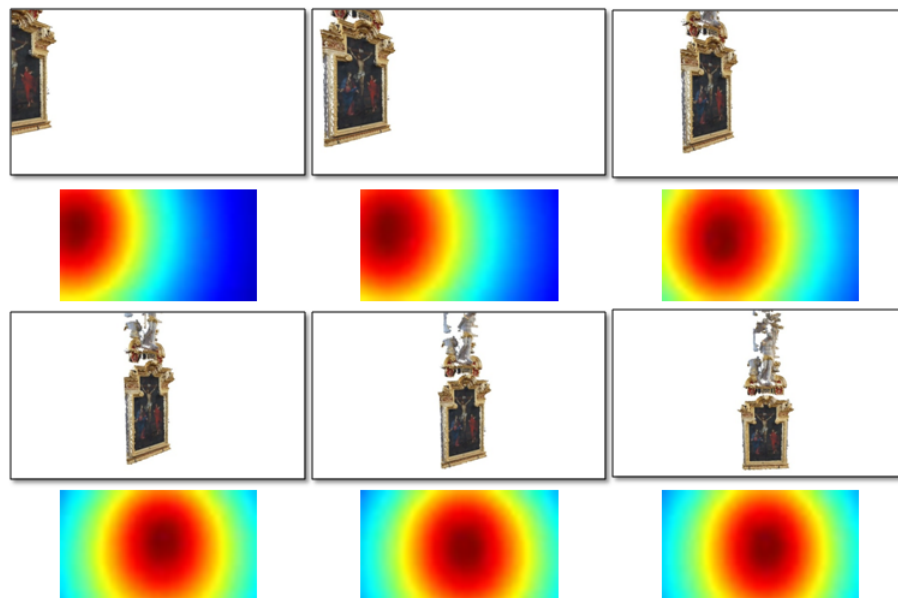


FIGURE 3.28 – Exemple 1 : quelques images clés illustrant la tâche de cadrage visuel assisté et leurs cartes de mélange de gaussiennes correspondantes.

La figure 3.29 représente une vue de dessus du modèle 3D avec la trajectoire suivie par la caméra virtuelle pour l'accomplissement de la tâche de cadrage visuel.

Une autre expérience a été achevée, mais cette fois-ci en plaçant la caméra virtuelle à une autre pose initiale. Dans cet exemple, l'œuvre d'art est dès le départ centrée dans l'image rendue par rapport à sa largeur. C'est pour cela, que dans la carte initiale des mélanges de gaussiennes, le mode principal est également centré. Cependant, la caméra n'est pas bien orientée par rapport à l'œuvre d'art, la peinture

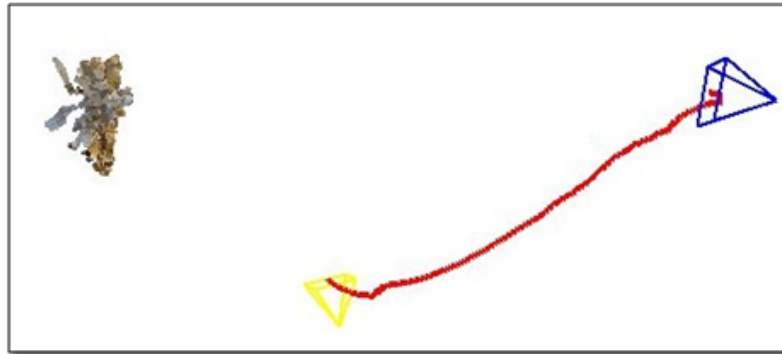


FIGURE 3.29 – Exemple 1 : vue de dessus du modèle avec la trajectoire de la caméra (la pyramide jaune correspond à la pose initiale de la caméra tandis que la bleue à la pose finale).

et la statue ne sont pas bien mises en valeur par ce point de vue et on ne voit qu'une partie de l'objet. C'est pourquoi, dans la carte du mélange de gaussiennes initial le mode principal est un peu décalé vers le haut. Le but est alors de corriger la position et surtout l'orientation de la caméra en prenant en compte l'information de pertinence. Ce qui est réalisé avec succès comme on peut observer sur les images de la figure 3.30 et la trajectoire prise par la caméra virtuelle dans la figure 3.31.

On observe à travers ces deux expériences que la surface du mélange de gaussiennes est uniforme sur la carte grâce à la méthode utilisée (Équation (3.43)) pour le calcul du paramètre λ .

3.4.4 Évaluation du système

Une évaluation basée sur l'avis d'un échantillon d'utilisateurs a été menée afin de qualifier le cadrage visuel dans les trois modes suivants :

- Le mode manuel : l'utilisateur utilise les touches du clavier afin de se déplacer librement dans l'environnement 3D.
- Le mode assisté : le déplacement de la caméra virtuelle est contrôlé par l'algorithme proposé du cadrage visuel assisté.
- Le mode automatique : la caméra est contrôlée directement par la méthode de cadrage automatique proposée.

L'objectif de cette expérience est de comparer ces trois modes quand la consigne donnée à l'utilisateur est de réaliser un cadrage "pertinent" selon son point de vue de tout l'objet 3D (l'œuvre d'art dans cet exemple). Cette évaluation a été suivie par dix participants (six hommes et quatre femmes) âgés de 23 à 50 ans. Des images de

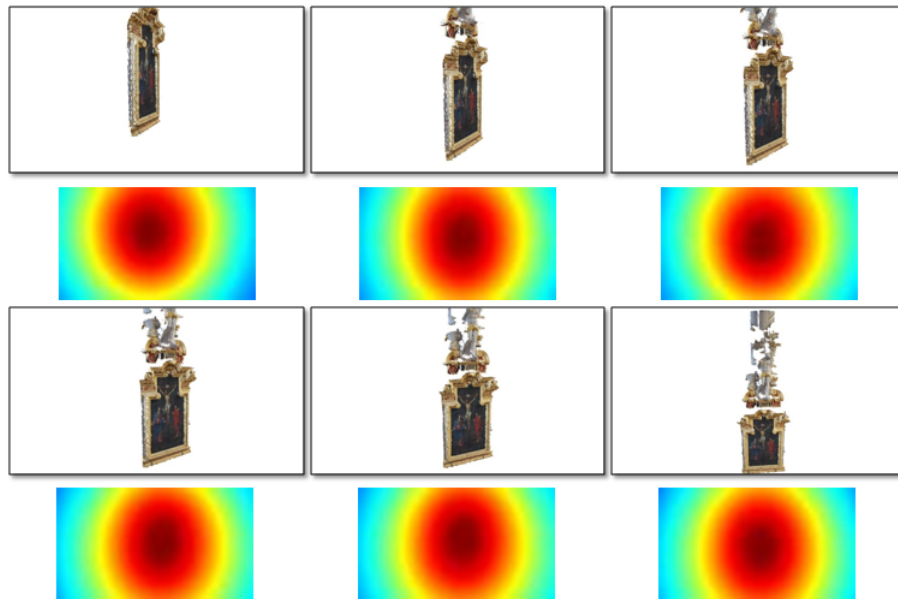


FIGURE 3.30 – Exemple 2 : quelques images clés illustrant la tâche de cadrage visuel assisté et leurs cartes de mélange de gaussiennes correspondantes.

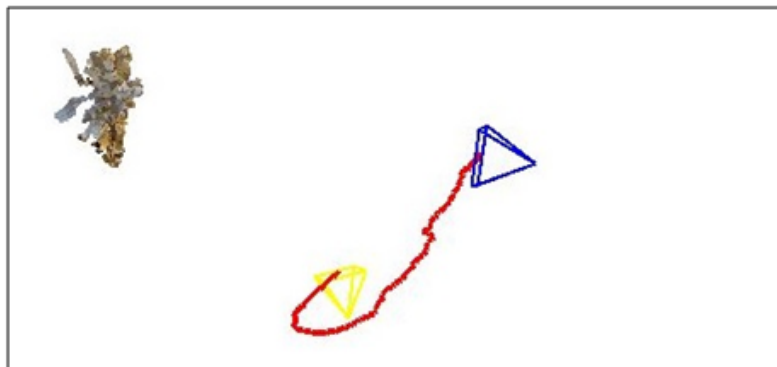


FIGURE 3.31 – Exemple 2 : vue de dessus du modèle avec la trajectoire de la caméra (la pyramide jaune correspond à la pose initiale de la caméra et la bleue à la pose finale).

résolution faible 60×30 sont utilisées pour le calcul de la loi de commande assistée, ce qui a permis d'accomplir la navigation en temps réel sans pour autant réduire la robustesse de l'algorithme.

L'utilisateur est devant une interface avec le modèle 3D de l'œuvre d'art rendu à partir d'une position initiale que nous avons définie. Ces conditions sont les mêmes

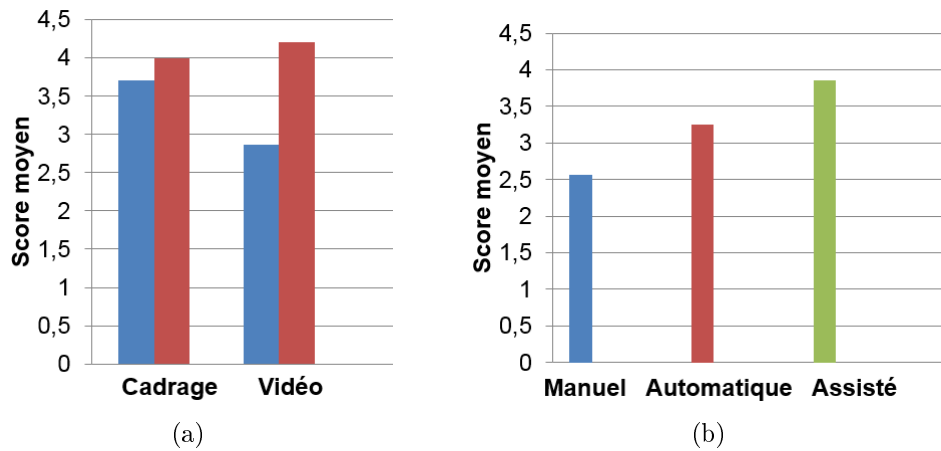


FIGURE 3.32 – (a) Evaluation de l'utilisateur concernant la qualité du cadrage et celle de la vidéo résultante (la colonne bleue correspond au mode manuel, tandis que la rouge au mode automatique); (b) Evaluation des utilisateurs des trois modes de navigation.

pour les dix utilisateurs. Le but est de réaliser la consigne en se déplaçant dans l'environnement à l'aide de quelques touches du clavier définies au préalable.

Les facteurs évalués sont la qualité du cadrage, qui correspond à l'image finale obtenue dans les deux modes automatique et manuel, ainsi que la qualité de la vidéo résultante (Figure 3.32a). L'utilisateur donne des notes comprises entre 0 et 5 pour évaluer le cadrage et la vidéo dans les deux modes. Les résultats obtenus sont : une note moyenne de 3.7 pour le mode manuel et 4 pour le mode automatique. On remarque que les utilisateurs sont plus intéressés par le résultat obtenu avec le mode automatique parce que dans ce mode la caméra réussit à bien se positionner en mettant en évidence le tableau et la statue tout en centrant l'œuvre d'art. Concernant la qualité de la vidéo, on obtient un score moyen de 2.8 pour le mode manuel et de 4.2 pour le mode automatique. Les utilisateurs ont beaucoup plus apprécié la vidéo obtenue à partir du mode automatique. Ce score élevé pour le mode automatique est dû au fait que les d.d.l de la caméra sont couplés ce qui assure un mouvement fluide et non saccadé, contrairement au mode manuel où les d.d.l ne sont pas couplés en raison de l'utilisation du clavier.

La figure 3.32b illustre une étude de satisfaction pour comparer les trois modes. Encore une fois les utilisateurs sont priés de donner des notes comprise entre 0 et 5 pour l'évaluation des trois modes. Les utilisateurs semblent davantage apprécier le mode assisté parce qu'il leur permet d'avoir plus de liberté tout en arrivant au résultat escompté.

3.5 Conclusion

Dans ce chapitre, dans un premier temps nous nous sommes intéressés à l'entropie photométrique. Il s'est avéré que son exploitation pour la résolution du problème de cadrage visuel est inappropriée. Sa modélisation basée sur l'équilibre de l'histogramme des niveaux de gris de l'image la rend dépendante de la nature des images utilisées pour le calcul de la loi de commande. Quand les niveaux de gris de l'histogramme ne sont pas bien équilibrés dans l'image, l'algorithme d'optimisation converge vers des maximums locaux et ainsi échoue comme ceci est constaté expérimentalement en simulation. Cependant, quand les niveaux de gris de l'histogramme sont équilibrés, comme c'est le cas dans l'expérimentation réalisée dans le nuage de points colorés d'un environnement réel, la caméra se déplace en maximisant l'information dans l'image mais sans réussir la tâche de cadrage visuel. C'est pourquoi, nous avons proposé dans le Chapitre 4 d'utiliser l'entropie photométrique dans un contexte d'exploration à plus grande échelle en intégrant un mécanisme qui permet de passer outre les maximums locaux.

D'un autre côté, une étude a été réalisée afin de mettre en avant la nouvelle représentation des mélanges de gaussiennes basées sur la saillance photométrique. En effet, on a pu démontrer qu'il était nécessaire d'utiliser un mélange de gaussiennes basées saillance dans la conception de la fonction de coût au lieu de la saillance directe. Cette nouvelle primitive est en parfait accord avec le problème du cadrage visuel en plus de son rôle dans l'amélioration de la convergence et de la robustesse du cadrage visuel. Les résultats ont montré que le problème des maximums locaux apparus avec l'utilisation de l'entropie photométrique est résolu. En plus de cela, dans des environnements réduits l'asservissement basé gaussiennes de saillance est temps-réel alors que celui basé entropie photométrique ne l'est pas. Différentes expérimentations avec un nuage de points 3D ainsi que dans un environnement réel avec une application robotique ont clairement permis de valider la méthode proposée et de conclure qu'elle est robuste et répond à nos attentes.

Enfin, le problème du contrôle du déplacement 3D d'une caméra virtuelle a été abordé afin de réaliser un cadrage visuel assisté d'un objet 3D. La nouvelle primitive visuelle : mélange de gaussiennes basées saillance a été utilisée dans ce problème combinée avec une contrainte de gestion de distance, et enfin couplée d'une manière adéquate avec l'interaction de l'utilisateur. Cette approche a été validée en simulation et sur l'œuvre d'art de la chapelle Saint Sébastien de la cathédrale d'Amiens. Les résultats ont montré que la méthode est robuste malgré l'utilisation d'images de faible résolution pour le calcul. Le cadrage visuel fonctionne en temps

réel car l'utilisation d'une résolution basse diminue considérablement le temps de calcul. Une étude préliminaire basée sur l'avis d'utilisateurs a été menée et a montré l'intérêt de notre contribution.

Exploration virtuelle d'un environnement 3D

Sommaire

4.1	Exploration automatique basée entropie visuelle	89
4.1.1	Formalisation du problème : tâche et contraintes	89
4.1.2	Schéma de résolution : hybride et hiérarchique	92
4.1.3	Résultats expérimentaux	97
4.1.4	Conclusion	101
4.2	Exploration assistée : mélanges de gaussiennes basées saillance	102
4.2.1	Formalisation sous forme de contraintes	104
4.2.2	Schéma de résolution : redondance et hiérarchie	107
4.2.3	Résultats expérimentaux	111
4.2.4	Conclusion	114

La manipulation des interfaces d'interaction accompagnant les scanners laser est une tâche plutôt difficile pour un utilisateur novice, surtout quand le but est de naviguer dans un environnement de grande taille, complexe et constitué d'un nuage de points 3D. La liberté que procurent ces outils engendre plusieurs difficultés comme la gestion de l'orientation de la caméra virtuelle, l'accès à des zones spécifiques dans le modèle pour réussir à visualiser un rendu réaliste et pertinent. L'objectif dans ce chapitre est de proposer un outil d'assistance à la navigation en prenant en charge la gestion de plusieurs d.d.l de la caméra virtuelle. L'utilisateur, s'il le souhaite, peut être aidé par un système de contrôle automatique ou semi-automatique qui va l'aider dans sa tâche d'exploration. Ce problème d'assistance a été traité de deux manières différentes. La première en fournissant à l'utilisateur une liberté très réduite tandis que dans la seconde, plus de liberté lui est attribuée.

4.1 Exploration automatique basée entropie visuelle

Dans cette première approche de l'assistance, l'utilisateur définit uniquement une position initiale de la caméra et le système, que nous allons exposer, guidera automatiquement la caméra de façon à générer à la fois des mouvements pertinents et réalistes. Ce problème est modélisé sous la forme d'une tâche principale et de contraintes. La tâche principale décrit la pertinence visuelle, qui est basée sur l'hypothèse que l'utilisateur est toujours attiré par les endroits riches en information visuelle. Cette formulation s'appuie sur la théorie de l'information et plus précisément sur l'entropie de Shannon. Par la suite, afin de garantir un mouvement réaliste, deux contraintes ont été ajoutées : l'évitement d'obstacles et la régularisation du mouvement dans l'image. La première contrainte assure un déplacement sécurisé de la caméra dans le modèle 3D, c'est-à-dire sans heurter ni traverser les obstacles. Tandis que la seconde contrainte, basée sur l'utilisation du flot optique, permet d'éviter les mouvements saccadés et les sauts brusques de la caméra. Un algorithme d'optimisation hiérarchique en deux étapes, qui permet de combiner la tâche et les contraintes en un seul problème et de le résoudre est proposé. La première étape consiste à déterminer la direction du mouvement. Ensuite, dans une seconde étape, l'amplitude du mouvement est régulée afin de garantir un mouvement dynamiquement pertinent. Dans le but de préserver un mouvement réaliste de la caméra, un mécanisme qui permet de commander cinq d.d.l opérationnels de la caméra (les trois translations, le « pan » et le « tilt ») afin d'éviter la rotation autour de l'axe optique a été intégré. Notre approche a été testée en navigation intérieure et extérieure sur deux modèles 3D très complexes : un environnement urbain et la cathédrale d'Amiens. Ces modèles 3D sont issus de mesures d'environnements réels par des relevés laser et images.

4.1.1 Formalisation du problème : tâche et contraintes

4.1.1.1 Vue d'ensemble

Comme dans le Chapitre 3, l'entropie visuelle photométrique est calculée à partir de l'intensité des pixels de l'image. Ceci la rend dépendante du point de vue et donc de l'image rendue et non du type du modèle 3D (nuage de points, maillage). Le contrôle automatique de la caméra est mis en œuvre par un asservissement visuel virtuel, en utilisant comme primitive l'entropie photométrique. En concevant la loi de commande qui maximise l'entropie, cette tâche permet de calculer la direction du mouvement sans aucune étape de planification ou de connexion entre des points de

vue choisis au préalable. Pour garantir des trajectoires réalistes, la caméra virtuelle doit éviter les obstacles durant son mouvement. Beaucoup de travaux en contrôle de caméra se sont intéressés à cette problématique comme les méthodes de lancer de rayons [Tomlinson 2000][Bourne 2008][Bares 1998], les méthodes de volumes englobants [Courty 2001] [Feddema 1989] et les méthodes basées image [Marchand 1998]. En ce qui nous concerne, la méthode proposée par Courty et al. [Courty 2001] qui consiste à maximiser une fonction de coût qui tend vers l'infini quand la distance entre la caméra et l'obstacle est nulle a été adaptée. Dans notre approche, la distance moyenne entre la caméra et tous les sommets du modèle qui se trouvent dans une sphère, à une distance fixe et centrée sur la caméra virtuelle, est maximisée.

La maximisation de la tâche principale, l'entropie photométrique, tout en respectant l'évitement d'obstacles (la première contrainte) sont combinées en utilisant une loi de contrôle hybride qui consiste à empiler les deux tâches dans un même système en les pondérant [Hager 1997]. Cela permet d'éviter les changements de direction soudains à chaque fois qu'un obstacle est détecté. Quand la loi de contrôle basée sur l'entropie photométrique et l'évitement d'obstacles converge, la pertinence devient quasiment constante (la même quantité d'information dans l'image). Cependant, pour que le mouvement soit dynamiquement pertinent, les sauts brusques entre deux images successives (discontinuité du mouvement dans l'image) doivent être évités. Pour résoudre ce problème de discontinuité du mouvement dans l'image, une loi de contrôle basée sur le flot optique est proposée, destinée à régulariser le mouvement dans l'image.

Pour prendre en compte la maximisation de l'entropie, l'évitement des obstacles et la régularisation du flot optique, une loi de contrôle hiérarchique en deux étapes est considérée, comme illustré sur la figure 4.1. Elle combine ces critères et estime la direction et l'amplitude du mouvement.

4.1.1.2 L'entropie photométrique et l'évitement d'obstacles

La même définition de l'entropie photométrique utilisée dans le chapitre 3 pour le cadrage visuel va être exploitée (Équation 3.2). Nous rappelons que la probabilité est exprimée à l'aide d'une B-spline (Équation 3.12), qui représente une approximation de l'équilibre des niveaux de gris de l'histogramme dans l'image.

De la même façon aussi que dans le chapitre 3, la fonction de coût qui permet d'éviter les obstacles (Équation 3.53) tend vers l'infini quand la distance entre la caméra et l'obstacle est nulle.

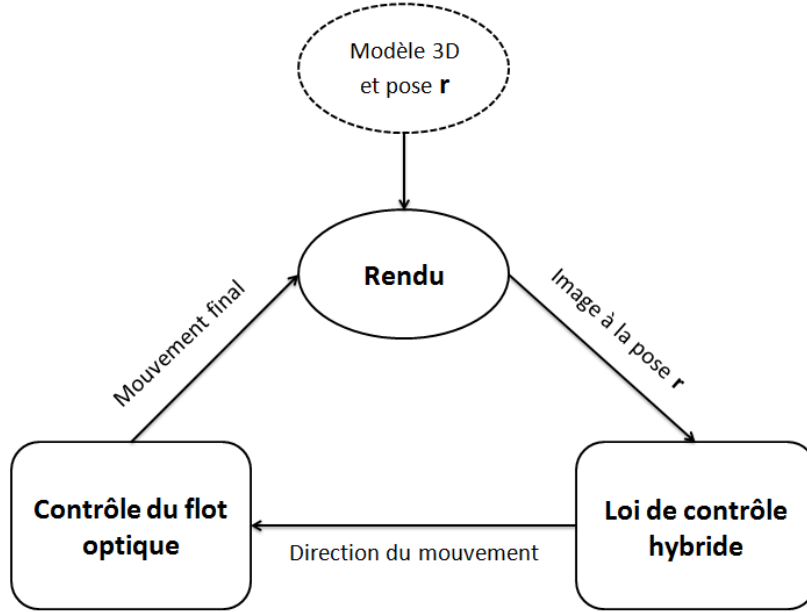


FIGURE 4.1 – La stratégie hiérarchique.

4.1.1.3 Contrainte basée sur le flot optique

Cette contrainte a pour rôle d'éviter les sauts brusques de la caméra lors de son déplacement. La cause principale de ce problème est relative aux algorithmes d'optimisation non-linéaire qui fournissent des incréments de pose variables. D'où la nécessité de réguler ces variations pour obtenir un rendu régulier. La régularisation du mouvement est faite dans l'image pour assurer un mouvement fluide du rendu. L'approche consiste à fixer une amplitude du flot optique entre l'image courante et l'image résultante de la loi de contrôle hybride qui combine la maximisation de l'entropie photométrique et l'évitement des obstacles. La direction du mouvement est préservée comme on peut remarquer sur la figure 4.2 alors que son amplitude est corrigée en la fixant empiriquement.

L'amplitude fixée pour le flot optique est exprimée de la manière suivante :

$$\Delta \mathbf{u}_{des} = \left[\Delta u_{des}, \Delta v_{des} \right]^T \quad (4.1)$$

Ensuite l'équation de conservation des intensités est utilisée pour lier l'amplitude du flot désiré $\Delta \mathbf{u}_{des}$ et la pose de la caméra \mathbf{r} :

$$I(\mathbf{u}, \mathbf{r}) = I(\mathbf{u} + \Delta \mathbf{u}_{des}, \mathbf{r} + \dot{\mathbf{r}}_{kf}) \quad (4.2)$$

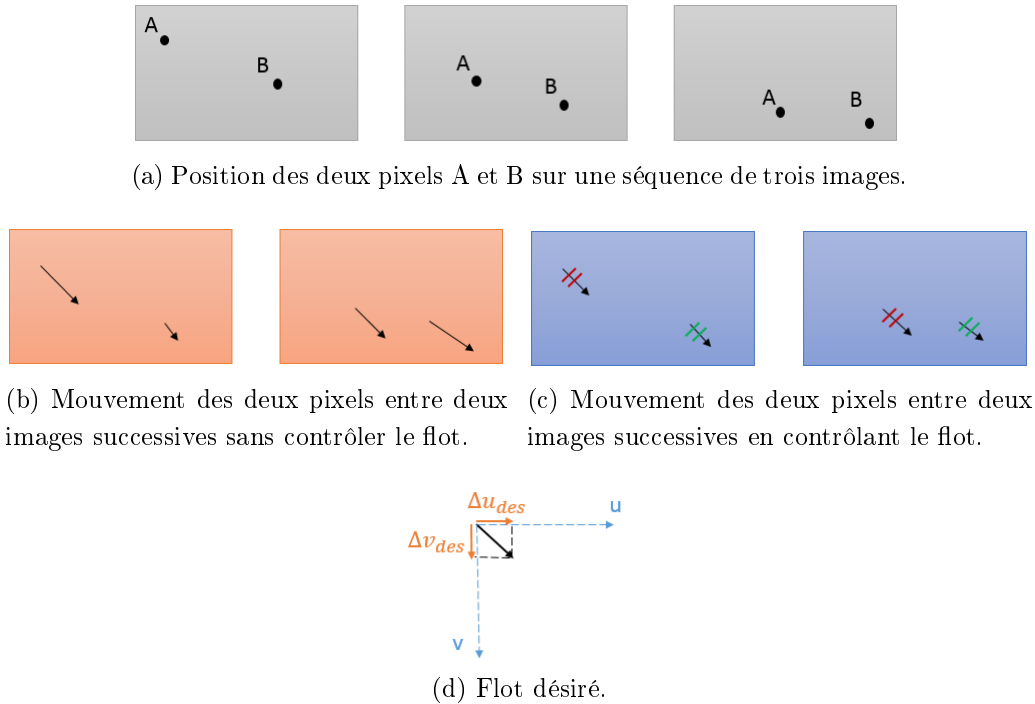


FIGURE 4.2 – Exemple de régularisation de l'amplitude du mouvement dans l'image de deux pixels A et B.

où \mathbf{r}_{kf} est l'incrément de pose final qui permettra de prendre en compte cette dernière contrainte.

4.1.2 Schéma de résolution : hybride et hiérarchique

La tâche principale et les contraintes sont non linéaires par rapport à la pose de la caméra \mathbf{r} . Un processus de résolution non linéaire itératif est appliqué. La linéarisation est effectuée à l'aide du développement de Taylor. Le cas général d'une loi de contrôle pour les six d.d.l cartésiens de la caméra est d'abord considéré. Ensuite la restriction à cinq degrés opérationnels est présentée à la section 4.1.2.5. Le processus itératif consiste à estimer l'incrément de pose \mathbf{r}_k afin de modifier la pose courante \mathbf{r}_{k-1} (Équation (3.13)).

4.1.2.1 Critère de l'entropie photométrique

Nous rappelons que la maximisation de l'entropie photométrique est équivalente à la minimisation de son opposé $-E(\mathbf{r}_k)$. Ainsi, la recherche du maximum revient à annuler la dérivée de l'entropie photométrique (Équation 3.15). Pour le calcul de

la loi de contrôle basée entropie, il est nécessaire de calculer la Jacobienne et le Hessien. Leurs expressions sont données dans le Chapitre 3 respectivement par les équations 3.17 et 3.23.

4.1.2.2 Contrainte d'évitement d'obstacles

Le but ici est de minimiser l'équation (3.53). l'algorithme de descente du gradient est exploité pour la résolution. Le but est de garantir à la caméra virtuelle d'éviter les obstacles (les sommets du modèle 3D) durant sa navigation dans le modèle 3D. De ce fait, seule la position de la caméra est considérée et non son orientation.

4.1.2.3 Contrainte basée sur le flot optique

Afin de régulariser l'amplitude du flot optique dans l'image, l'équation de conservation du flot optique (Équation 4.2) est considérée. On lui applique le développement de Taylor au 1^{er} ordre :

$$I(\mathbf{u} + \Delta\mathbf{u}_{des}, \mathbf{r}_k + \mathbf{r}_{k_f}^i) = I(\mathbf{u}, \mathbf{r}_k) + \frac{\partial I}{\partial \mathbf{u}} \Delta\mathbf{u}_{des} + \frac{\partial I}{\partial \mathbf{r}} \mathbf{r}_{k_f}^i \quad (4.3)$$

tel $\mathbf{r}_{k_f}^i$ est l'incrément de pose final qui prendra en compte la contrainte de régularisation du flot en plus de la loi de contrôle hybride.

$$\frac{\partial I}{\partial \mathbf{u}} \Delta\mathbf{u}_{des} + \frac{\partial I}{\partial \mathbf{u}} \mathbf{L}_u \mathbf{r}_{k_f}^i = 0 \quad (4.4)$$

Cette dernière équation ne tient compte que des éléments connus, notamment, le gradient de l'image et la matrice d'interaction.

4.1.2.4 Algorithme hiérarchique

Pour combiner la tâche principale (maximisation de l'entropie photométrique) et les autres contraintes, on pourrait faire une résolution globale en empilant toutes les équations dans une seule matrice. Cependant, le résultat obtenu n'est pas satisfaisant parce que les contraintes ne sont toujours pas toutes respectées. En effet, si on a plusieurs contraintes, une ou deux contraintes peuvent prendre le dessus sur les autres restantes. Ainsi, l'incrément de pose calculé ne tiendra pas compte de toutes les contraintes. Cette méthode a donc des limitations par rapport au nombre de contraintes et ne semble être efficace que si ce dernier est faible.

Les approches basées sur le formalisme de redondance et qui sont souvent utilisées pour faire du multi-tâches, permettent d'éviter les inconvénients de l'empilement des tâches. Cette technique est réalisée en affectant à chaque tâche un certain

nombre de d.d.l. Dans le formalisme de redondance, il n'est pas possible qu'un d.d.l soit exploité par deux tâches différentes. Il ne peut donc pas être exploité dans notre cas car la tâche principale requiert tous les d.d.l de la caméra. C'est pourquoi, nous avons proposé un nouvel algorithme hiérarchique de contrôle de caméra.

Ensuite, l'amplitude du mouvement est ajustée afin de générer un flot optique globalement constant dans le but d'avoir un mouvement régulier dans l'image. La trajectoire de la caméra est générée à l'aide d'un processus d'optimisation itératif qui prend en compte ces deux étapes.

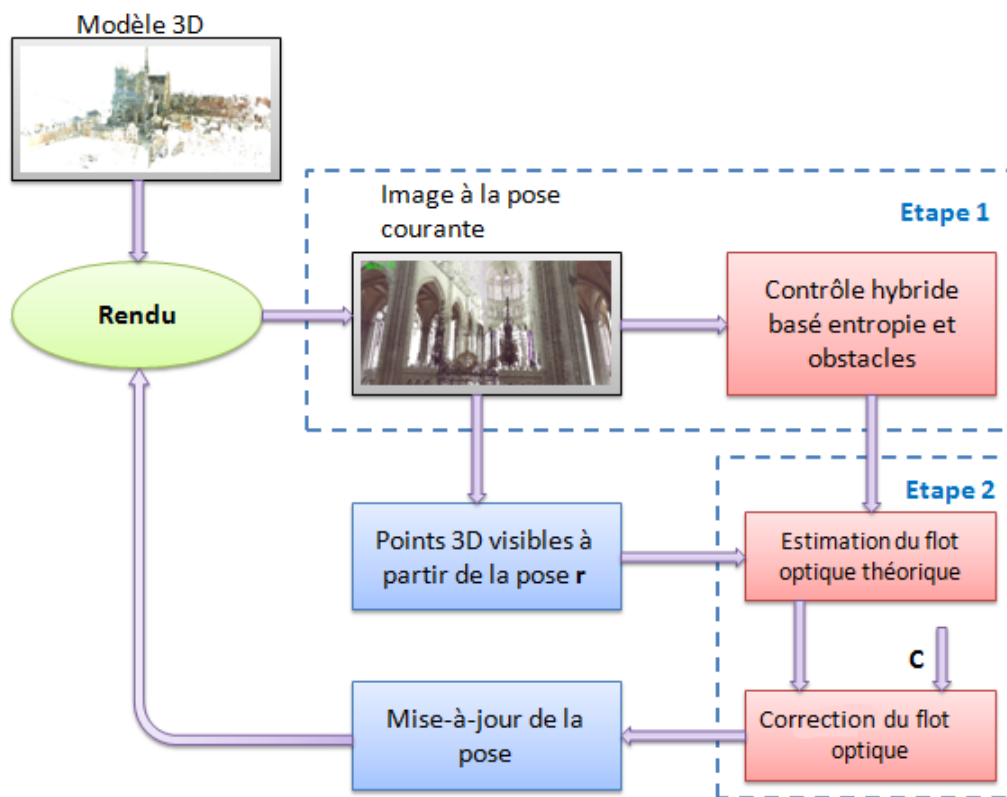


FIGURE 4.3 – Processus de contrôle de caméra.

Étape 1 : estimation de la direction du mouvement : Elle est obtenue en empilant directement les équations qui font référence à la maximisation de l'entropie et à l'évitement des obstacles (Équations (3.29) et (3.56)). L'estimation de

l'incrément hybride \mathbf{r}_k est donnée par :

$$\mathbf{r}_k = \begin{pmatrix} d_{\mathbf{r}}^2(-E(\mathbf{r}_{k-1})) \\ d_{\mathbf{r}}(O(\mathbf{r}_{k-1})) \end{pmatrix}^+ \begin{pmatrix} d_{\mathbf{r}}(-E(\mathbf{r}_{k-1})) \\ -O(\mathbf{r}_{k-1}) \end{pmatrix} \quad (4.5)$$

Dans ce qui suit on pose :

$$\mathbf{H}^+ = \begin{pmatrix} \frac{\partial^2(-E(\mathbf{r}_{k-1}))}{\partial \mathbf{r}^2} \\ \frac{\partial(O(\mathbf{r}_{k-1}))}{\partial \mathbf{r}} \end{pmatrix}^+ \quad (4.6)$$

et,

$$\mathbf{g} = \begin{pmatrix} \frac{\partial}{\partial \mathbf{r}}(-E(\mathbf{r}_{k-1})) \\ -O(\mathbf{r}_{k-1}) \end{pmatrix} \quad (4.7)$$

Étape 2 : Estimation finale de la pose Cette étape permet de borner l'amplitude du mouvement en procédant dans l'image en deux temps :

1. Le flot optique théorique Δu et Δv est calculé comme illustré sur la Figure 4.4, où $I(\mathbf{r}_{k-1})$ et $I(\mathbf{r}_{k-1} + \mathbf{r}_k)$ sont les images de synthèse rendues par le moteur graphique à partir de la pose courante et la pose résultante de l'étape 1. La matrice de transformation étant connue entre les deux poses, le flot optique théorique est calculé directement à partir de la différence entre les coordonnées des pixels des deux images rendues.

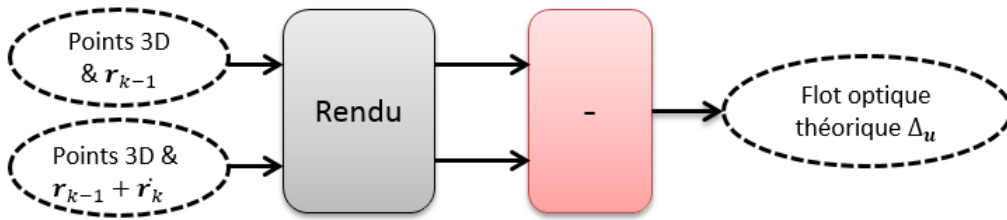


FIGURE 4.4 – Calcul du flot optique théorique (calcul de la différence entre les coordonnées des points des images rendues à la pose \mathbf{r}_{k-1} et $\mathbf{r}_{k-1} + \mathbf{r}_k$).

2. On fixe une constante C comme amplitude moyenne désirée du flot optique global. Ensuite, on impose cette amplitude en chaque pixel :

$$\Delta u_{des} = \frac{C \cdot \Delta u}{\sqrt{\Delta u^2 + \Delta v^2}} \quad (4.8)$$

$$\Delta v_{des} = \frac{C \cdot \Delta v}{\sqrt{\Delta u^2 + \Delta v^2}} \quad (4.9)$$

Le déplacement final de la caméra est calculé à partir de ce flot désiré. Pour cela, on effectue un développement de Taylor au premier ordre sur l'équation de conservation des intensités des pixels (Équation (4.4)) :

$$\frac{\partial I}{\partial \mathbf{r}} \dot{\mathbf{r}}_{k_f} = -\frac{\partial I}{\partial u} \Delta u_{des} - \frac{\partial I}{\partial v} \Delta v_{des} \quad (4.10)$$

$$\frac{\partial I}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \mathbf{r}} \dot{\mathbf{r}}_{k_f} = -\frac{\partial I}{\partial u} \Delta u_{des} - \frac{\partial I}{\partial v} \Delta v_{des} \quad (4.11)$$

$$\begin{bmatrix} \frac{\partial I}{\partial u} & \frac{\partial I}{\partial v} \end{bmatrix} \mathbf{L}_u \dot{\mathbf{r}}_{k_f} = - \begin{bmatrix} \frac{\partial I}{\partial u} & \frac{\partial I}{\partial v} \end{bmatrix} \begin{bmatrix} \Delta u_{des} \\ \Delta v_{des} \end{bmatrix} \quad (4.12)$$

Ainsi, la loi de contrôle finale peut directement être exprimée :

$$\dot{\mathbf{r}}_{k_f} = -\mathbf{L}_u^+ \Delta \mathbf{u}_{des} \quad (4.13)$$

où \mathbf{L}_u^+ est la pseudo inverse de la matrice d'interaction \mathbf{L}_u .

Comme précédemment mentionné, cette limitation ne modifie pas la direction du mouvement 3D, mais uniquement son amplitude.

4.1.2.5 Gestion des rotations

La même démarche que le chapitre précédent en restreignant la caméra à cinq d.d.l opérationnels est utilisée.

Ainsi, l'incrément de pose calculé par l'équation (4.5) devient :

$$\dot{\mathbf{q}}_k = (\mathbf{H} \ ^c\mathbf{J}_c)^+ \mathbf{g} \quad (4.14)$$

où $\ ^c\mathbf{J}_c$ le Jacobien dans le repère caméra. Le même raisonnement est appliqué pour la régularisation du flot (Équation (4.13)) :

$$\dot{\mathbf{q}}_{k_f} = (\mathbf{L}_u \ ^c\mathbf{J}_c)^+ \Delta \mathbf{u}_{des} \quad (4.15)$$

$\dot{\mathbf{q}}_{k_f}$ représente le vecteur à cinq dimensions de l'incrément de pose finale.

4.1.2.6 Gestion des optimums locaux

Une alternance entre le processus de maximisation et de minimisation de l'entropie a été implémentée pour éviter de stagner dans un maximum local. On commence par le processus de maximisation de l'entropie photométrique pour amener

la caméra vers les parties les plus pertinentes du modèle, ensuite un processus de minimisation de l'entropie est déclenché pour changer de zone. Le flot optique est borné afin de passer plus de temps à contempler les informations pertinentes de la scène durant l'étape de maximisation. A l'inverse, durant l'étape de minimisation, les mouvements lents ne sont pas pertinents car l'information visuelle devient de plus en plus pauvre. Pour cela, le flot optique est régulé durant cette étape pour avoir un mouvement à vitesse constante. Cette alternance permet de générer une plus grande trajectoire. La durée de chaque étape peut être définie par l'utilisateur.

4.1.3 Résultats expérimentaux

L'approche proposée a été testée sur deux modèles 3D composés de nuages de points colorés de modèles très complexes à large échelle. L'approche est validée dans un environnement urbain et un modèle partiel de la cathédrale d'Amiens. Ces environnements sont obtenus par relevés laser (Faro Focus 3D et Leica). Comme illustrent les images rendues, l'environnement n'a pas été scanné dans sa totalité (zones blanches dans les images). Chaque environnement utilisé, compte 100 millions de points 3D. Pour la visualisation des modèles, le moteur graphique Ogre 3D [Ogr] est utilisé.

4.1.3.1 Maximisation de l'entropie photométrique

Dans cette expérience, le comportement de la maximisation de l'entropie photométrique est mis en avant. La maximisation de l'entropie est appliquée en navigation extérieure sur le modèle de la cathédrale. A partir d'une position initiale où le modèle 3D ne remplit pas totalement l'image (Figure 4.5), la loi de contrôle basée entropie photométrique contraint la caméra à inclure le plus d'informations dans son champ de vue. La Figure 4.6 montre l'évolution de la fonction de coût tout au long de la trajectoire.

4.1.3.2 Évitement d'obstacles

La contrainte d'évitement d'obstacles est obligatoire en navigation virtuelle car elle permet de gérer les comportements irréalistes de la caméra virtuelle, comme traverser des murs ou des piliers à l'intérieur du modèle de la cathédrale. La Figure 4.7 montre le comportement de la caméra suite à la loi de contrôle hybride entropie-obstacles (Équation (4.5)). La Figure 4.7a illustre un extrait des images de synthèse à chaque point de vue de la caméra, tandis que la Figure 4.7b représente les images équirectangulaires correspondantes contenant les obstacles (les sommets)

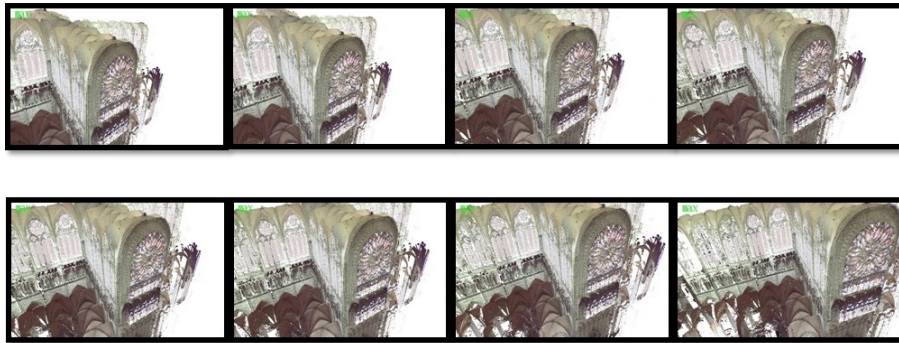


FIGURE 4.5 – Navigation automatique en maximisant l'entropie photométrique : vue d'extérieur du modèle.

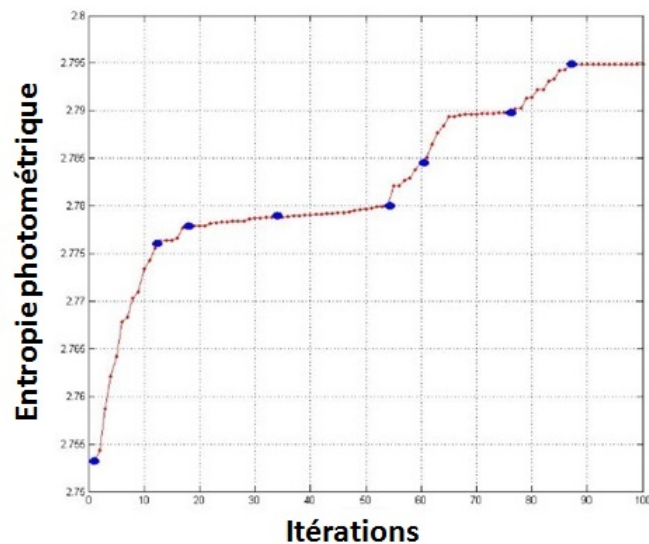


FIGURE 4.6 – Évolution de l'entropie photométrique. les points bleus correspondent aux images de la Figure 4.5 (de gauche à droite et de haut en bas).

déTECTÉS à cette pose. On remarque, qu'à partir de la quatrième image, la caméra s'éloigne des obstacles, ainsi le nombre de sommets dans l'image équirectangulaire diminue.

Dans la figure 4.8a, on observe en vert la trajectoire de la caméra projetée sur le plan 2D du modèle 3D de la cathédrale. Cette trajectoire montre que la caméra passe entre deux piliers pour éviter les collisions. La Figure 4.8b montre l'évolution de la fonction de coût tout au long du mouvement de la caméra. On remarque que la fonction de coût atteint son maximum quand le nombre d'obstacles détectés dans



(a) Séquence d'images en navigation intérieure.



(b) Obstacles (sommets) détectés à partir de chaque pose qui a permis de générer les images de synthèses.

FIGURE 4.7 – Comportement de la contrainte d'évitement d'obstacles.

la sphère augmente. Cela correspond à la troisième image de la Figure 4.7b.

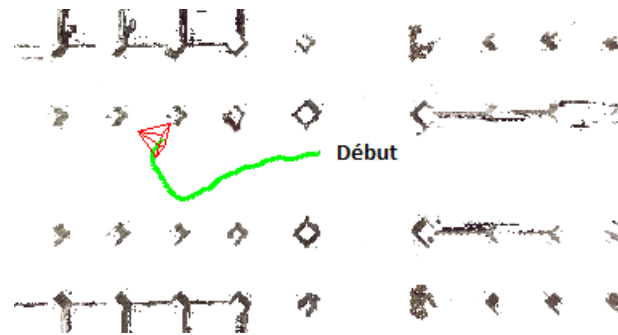
4.1.3.3 Contrôle du flot optique

Dans cette expérience, la caméra est contrôlée en utilisant l'algorithme hiérarchique. Dans le but de mettre en évidence l'importance de cette contrainte, nous avons, à partir d'une même pose initiale, appliqué la loi de contrôle avec et sans la contrainte basée sur le flot optique. La figure 4.9 qui montre l'évolution de la moyenne de la norme du flot optique calculée pour les deux séquences, met en évidence que cette contrainte est primordiale dans la mesure où elle permet de lisser le mouvement et d'éviter des sauts brusques. La courbe avec contrôle du flot montre qu'un flot moyen régulier est maintenu en comparaison avec la courbe sans contrôle du flot.

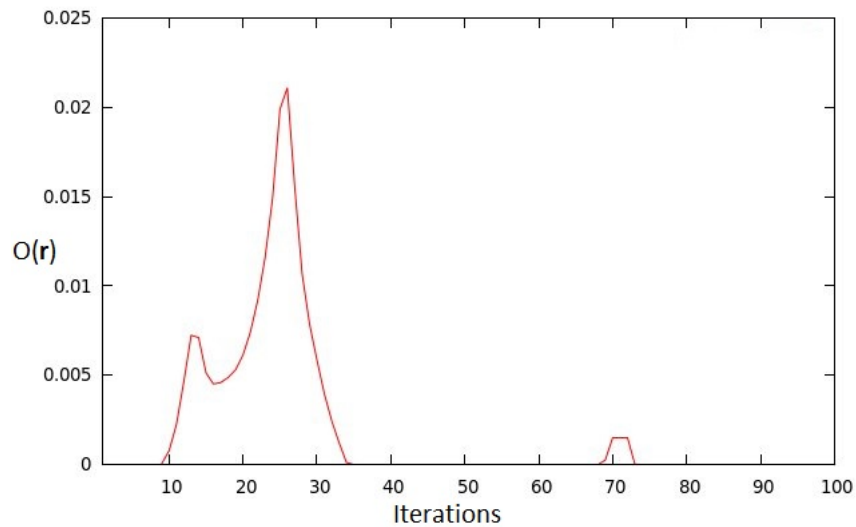
4.1.3.4 Navigation réaliste

Ce qui est décrit précédemment permet d'atteindre une zone dans le modèle à maximum d'entropie photométrique. Dans cette section, le but est de réaliser une navigation de longue durée. Ainsi, nous avons proposé d'alterner la maximisation et la minimisation de l'entropie.

La Figure 4.10 montre des résultats avec un environnement extérieur, qui a été choisi parce qu'il est plus simple que la cathédrale; il permet de mieux évaluer qualitativement la méthode. Cette montre figure quelques images extraites de la vidéo générée en appliquant l'algorithme proposé. En commençant à partir d'une



(a) Trajectoire de la caméra sur le plan 2D du modèle 3D de la cathédrale.



(b) Evolution de la fonction de coût d'évitement d'obstacles.

FIGURE 4.8 – Trajectoire de la caméra.

pose qui contient une partie de la façade, l'algorithme guide la caméra afin d'inclure tout le bâtiment dans son champ de vue.

Un résultat en navigation intérieure dans le modèle de la cathédrale, réalisé à l'aide du scanner laser Faro est présenté dans la Figure 4.11. Dans cette séquence, la caméra est initialement dirigée vers le cœur, ensuite un travelling combiné avec un panorama horizontal est réalisé pour explorer une partie de la croisée du transept.

Dans la Figure 4.12, un nouveau résultat de navigation automatique est illustré dans le modèle de la cathédrale, réalisé cette fois-ci à l'aide du scanner laser Leica. La caméra est initialement positionnée en face des arcs, ensuite elle recule tout en faisant un panorama horizontal en incluant le portail dans son champ de vue. A partir de là elle passe entre deux piliers pour se focaliser sur un seul pilier et recule

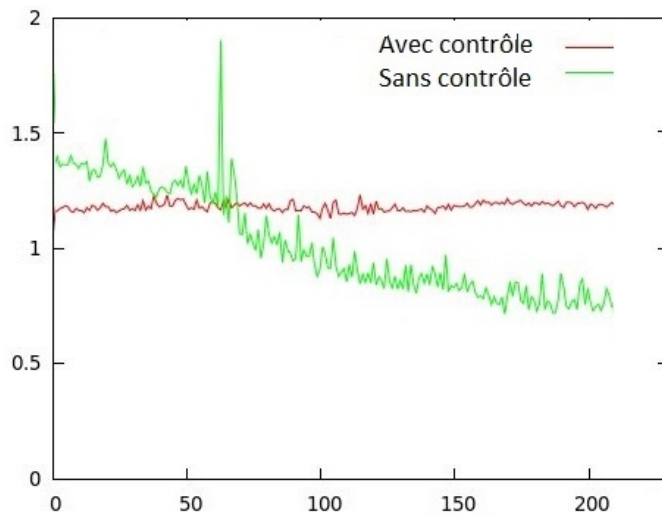


FIGURE 4.9 – Moyenne de la norme du flot optique pour une séquence d’images considérant la contrainte basée sur le flot (rouge) et une autre sans le considérer (vert).

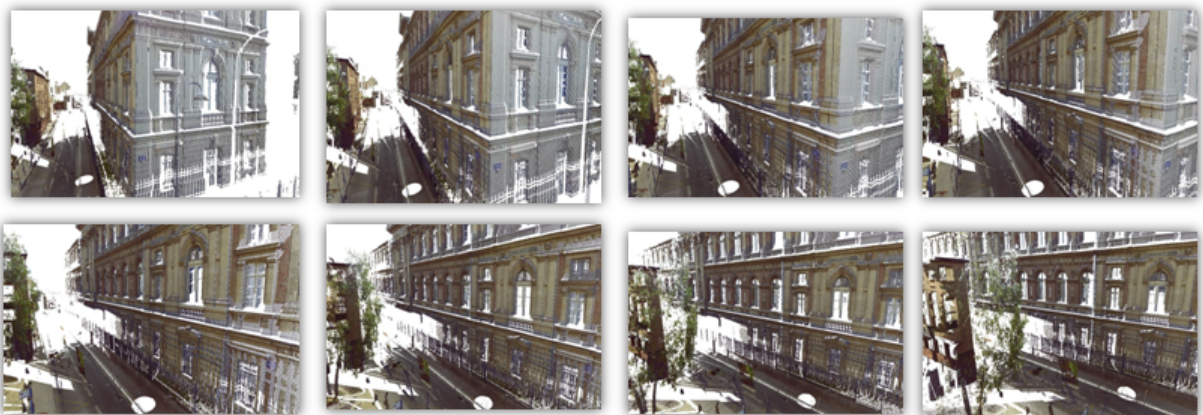


FIGURE 4.10 – Navigation automatique en extérieur.

pour visualiser tout le couloir avec les différents piliers et le second portail.

4.1.4 Conclusion

Nous avons proposé une méthode de contrôle automatique de caméra dans un environnement virtuel constitué d’un nuage de points 3D colorés. L’approche proposée

permet de générer des mouvements fluides, réalistes et pertinents. Le contrôle a été réalisé à l'aide d'un algorithme hiérarchique en deux étapes. La direction du mouvement est d'abord déterminée en combinant la tâche de maximisation de l'entropie photométrique et la contrainte d'évitement d'obstacles. L'entropie photométrique a pu montrer son intérêt pour se diriger vers les endroits intéressants d'une scène donnée. Ensuite, un contrôle du flot optique dans l'image a été réalisé afin d'éviter les mouvements brusques de la caméra. Les résultats obtenus montrent qu'à partir d'une position initiale pauvre en information, l'algorithme guide la caméra vers des endroits plus intéressants. Cette méthode est générale dans la mesure où on peut intégrer d'autres critères à optimiser.

Dans la section suivante, une autre primitive de pertinence visuelle va être exploitée. L'algorithme de résolution sera différent. L'interaction de l'utilisateur dans le système sera également prise en compte.

4.2 Exploration assistée : mélanges de gaussiennes basées saillance

L'approche présentée dans cette section garantit plus de liberté à l'utilisateur dans le choix des zones qu'il souhaite visiter de l'environnement 3D. Le système d'aide à la navigation élaboré repose sur une stratégie de navigation plus globale. Le but est d'aider l'utilisateur à réaliser une exploration plus complète d'un nuage



FIGURE 4.11 – Deuxième exemple : séquences d'image de navigation intérieure.

4.2. Exploration assistée : mélanges de gaussiennes basées saillance 103

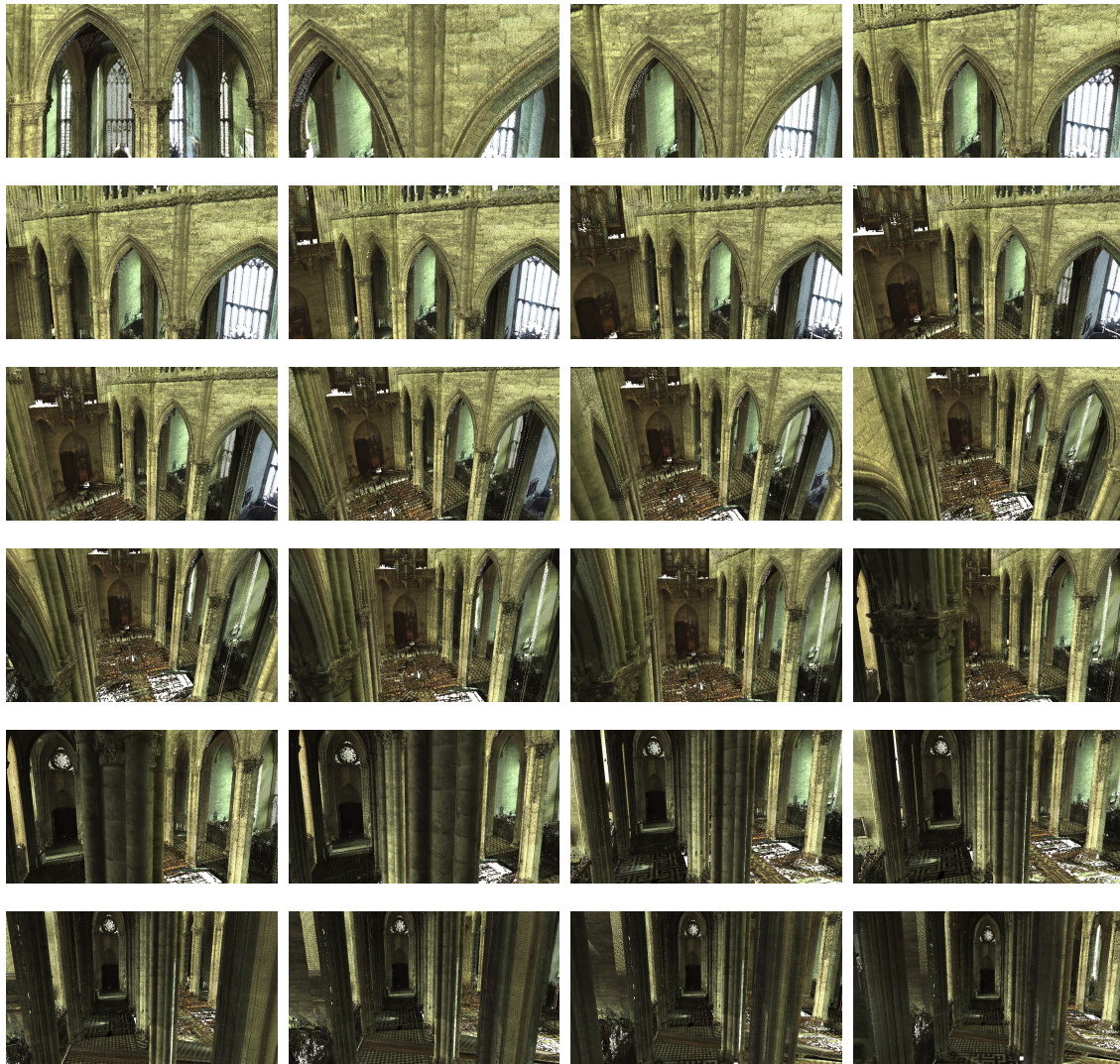


FIGURE 4.12 – Deuxième exemple : séquence d’images de navigation intérieure.

de points 3D de taille très significative et de complexité très élevée. Pour cela, la caméra virtuelle est pilotée à travers un ensemble de poses 3D (seules les positions (X, Y, Z) sont connues par exemple).

Les poses de la caméra à partir desquelles la visite est réalisée peuvent être calculées automatiquement à l’aide d’approches de planification de trajectoires [Andújar 2004, Lee 2005, Shtrom 2013], qui permettent également la construction d’une trajectoire passant par ces poses. De telles approches, qui sont automatiques, peuvent générer un nombre élevé de poses, si, potentiellement, de nombreuses parties de l’environnement, par exemple, sont pertinentes. La connexion entre ces poses peut engendrer plusieurs trajectoires, mais une seule doit être considérée pour réa-

liser l'exploration. Un choix qui peut s'avérer être difficile à faire par l'utilisateur pour la sélection de la trajectoire qui l'intéresse au plus. C'est pourquoi, il serait plus approprié d'inclure l'utilisateur dans le processus de définition de poses et de trajectoire.

Une autre façon de répondre à cette question, est d'assurer une plus forte implication de l'utilisateur dans le déroulement de l'exploration, définissant ainsi des poses manuellement dans les zones de l'environnement qu'il souhaite visiter. On peut imaginer facilement la définition d'un certain nombre de points de vue en les indiquant simplement, par exemple sur une carte 2D de l'environnement. L'enchaînement de ces points de passage aide à déterminer l'allure de la trajectoire pour l'approche la plus basique. L'ensemble de ces poses de caméra désirées est ensuite utilisé par une méthode automatique qui permet de générer une trajectoire en contrôlant les d.d.l de la caméra. Le calcul des trajectoires (poses intermédiaires de la caméra) qui lient deux points de vue définis par l'utilisateur peut être réalisé soit à l'aide d'une méthode d'interpolation, comme en animation traditionnelle [Lasseter 1987], ou en utilisant un asservissement visuel basé pose accompagné ou non de contraintes supplémentaires, comme l'évitement des obstacles et des occultations [Courty 2001, Bourne 2008, van Waveren 2008]. Cette dernière méthode permet de mettre à jour d'une manière itérative la pose courante de la caméra en minimisant sa distance par rapport à la pose désirée (point de vue désiré).

Mais un utilisateur qui découvre l'environnement à visiter, pourrait définir une orientation de la caméra très peu pertinente. Par conséquent, l'asservissement visuel basé pose peut produire des rendus de faible pertinence visuellement tout au long du mouvement de la caméra. La seule façon d'assurer la pertinence du rendu visuel est de prendre en considération le contenu de l'image. Ce dont l'asservissement basé pose ne tient pas compte. C'est la raison pour laquelle nous avons décidé d'exploiter la primitive de mélanges de gaussiennes basées saillance proposée dans le chapitre précédent. L'incorporation de cette contrainte avec l'asservissement basé pose est assurée à l'aide du formalisme de redondance [Rosen 1961, Chiaverini 1997]. Ceci est réalisé en attribuant les d.d.l de l'orientation de la caméra à la maximisation de la primitive mélange de gaussiennes basées saillance et les d.d.l de la position à l'asservissement basé pose.

4.2.1 Formalisation sous forme de contraintes

Dans le Chapitre 3, nous avons proposé une méthode de cadrage visuel efficace utilisant comme primitive photométrique le mélange de gaussiennes basées saillance.

4.2. Exploration assistée : mélanges de gaussiennes basées saillance 105

Cette méthode permet d'effectuer une navigation locale. Le but ici est d'étendre cette méthode pour effectuer une navigation à plus grande échelle. Pour illustrer ces approches, on peut considérer l'exemple de la visite d'un musée dans lequel on est à la fois intéressé par la découverte de chaque œuvre d'art, ce que permet le cadrage, et par l'exploration de toutes les pièces (Figure 4.13).

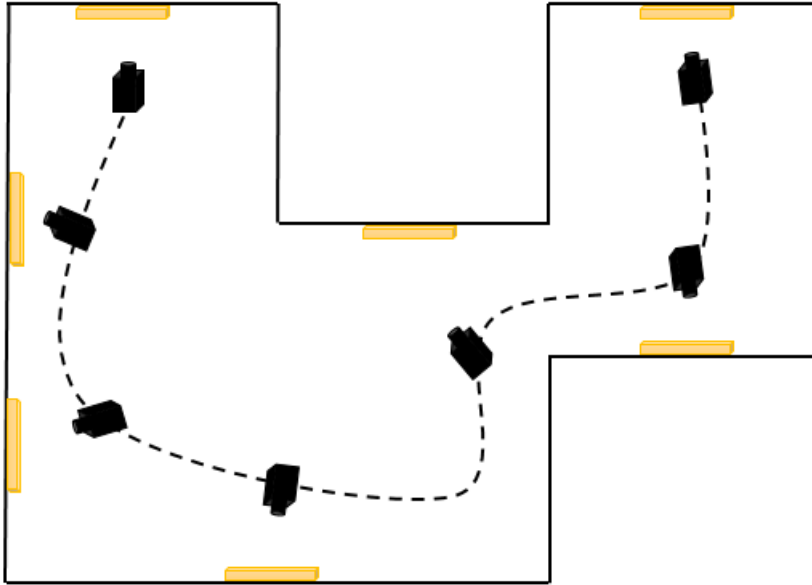


FIGURE 4.13 – Illustration de l'exploration d'un musée.

Étant donné un environnement composé d'un nuage de points 3D colorés (X, Y, Z, R, G, B) et un ensemble de positions cibles de la caméra définies par l'utilisateur éloignées les unes des autres \mathbf{p}_k (où $\mathbf{p}_k = (X, Y, Z) \forall k > 0$ et $\mathbf{p}_0 = (X_c, Y_c, Z_c, \theta_{wX}, \theta_{wY}, \theta_{wZ})$). Ces positions représentent les endroits que l'utilisateur souhaite visiter. L'idée du système d'aide à la navigation est d'utiliser cet ensemble de positions afin de construire automatiquement une trajectoire de la caméra passant par ces positions comme représenté dans la figure 4.13 en respectant toutes les contraintes suivantes :

- **Contrainte principale** : déplacer la caméra virtuelle d'une position initiale à une position finale. Pour ce faire, l'asservissement basé pose sera utilisé en contrôlant uniquement les trois translations afin de déplacer la caméra d'une position \mathbf{p}_k à une position \mathbf{p}_{k+1} .
- **Contrainte 1** : gestion de l'évitement des obstacles ;
- **Contrainte 2** : garder une orientation pertinente de caméra en maximisant le mélange de gaussiennes basées saillance et en ne contrôlant que les rotations

de la caméra afin de maintenir un contenu visuel pertinent.

- **Contrainte 3** : assurer des rotations réalistes de caméra (éviter la rotation de la caméra autour de l'axe optique en n'utilisant que ces cinq d.d.l opérationnels);
- **Contrainte 4** : garantir la pertinence dynamique en évitant les sauts soudains, les vitesses trop faibles ou trop élevées entre deux images rendues successivement durant le mouvement de la caméra. Afin de faire face à ce problème de discontinuité, particulièrement dans le mouvement apparent dans l'image, le flot optique est utilisé pour la régularisation du mouvement de la caméra.
- **Contrainte 5** : garder un contenu d'images non similaire, en d'autres termes, l'information contenue dans le champ de vue de la caméra doit être différente le long de la trajectoire de la caméra.

L'algorithme proposé, agit de la manière suivante. En premier lieu, l'asservissement visuel basé pose (**Contrainte principale**) et la contrainte d'évitement d'obstacles (**Contrainte 1**) sont combinés sous la forme d'une loi de contrôle hybride. Ceci permet d'éviter des changements soudains et répétitifs de la direction de la caméra virtuelle à chaque fois qu'un obstacle est détecté. Cette loi de contrôle hybride utilise les trois d.d.l de translations. Le formalisme de redondance est considéré afin de prendre en compte la maximisation du mélange de gaussiennes basées saillance (**Contrainte 2**) en n'utilisant que les d.d.l restants de rotations de la caméra. Dans les approches basées image, le formalisme de redondance a souvent été exploité afin de combiner l'exécution de plusieurs tâches. Cette technique est réalisée en attribuant à chaque tâche un nombre spécifique de d.d.l de la caméra. Une des caractéristiques distinctives de cette approche est qu'il est toujours question d'une tâche principale et d'autres secondaires. [Courty 2001] ont exploité cette méthode pour la combinaison de plusieurs tâches (suivi, évitement d'obstacles et d'occultations, ...).

La loi de contrôle basée redondance est combinée avec la contrainte du flot optique (**Contrainte 4**) en utilisant un schéma hiérarchique, dans lequel la direction du mouvement est obtenue à l'aide de la loi de contrôle basée redondance et son amplitude est obtenue par la contrainte de régularisation du flot optique. Cette contrainte est différente de celle présentée dans la section précédente, dans la mesure où elle consiste à borner le flot en le pondérant avec un facteur sans pour autant utiliser l'équation de conservation de l'intensité. Ainsi, le problème de discontinuité, en particulier dans l'apparence du mouvement dans l'image sera résolu.

```

Entrées : environnement 3D, ensemble des positions cibles  $S = \mathbf{p}_k$ , où ( $k \geq 0$ );
Résultat : Trajectoire de la caméra;
Initialisation :  $k = 0$ ;
Tant que ( $k \neq \text{taille}(S)$ ) faire
    Appliquer Contrainte principale, Contrainte 1, Contrainte 2,
Contrainte 3;
    Calculer la loi de contrôle basée redondance;
Si (contenu de l'image similaire) Alors
    | Appliquer Contrainte 5;
Sinon
    | Continuer;
Fin Si
    Appliquer Contrainte 4;
    Calculer la loi de contrôle hiérarchique;
Fait
Retourner resultat;
    
```

Algorithme 1 – Algorithme hiérarchique.

L'algorithme (Algorithm. 1) résume tout le processus pour calculer la loi de contrôle finale en considérant toutes les contraintes établies.

4.2.2 Schéma de résolution : redondance et hiérarchie

4.2.2.1 Loi de contrôle basée redondance

a- Contrôle des d.d.l de la caméra

Seuls les trois translations vont être utilisées pour le déplacement de la caméra virtuelle de la position courante \mathbf{p}_k à la position désirée \mathbf{p}_{k+1} tout en évitant les obstacles. Ceci va être accompli en respectant les étapes suivantes :

- Un asservissement visuel virtuel basé poses de la caméra est réalisé en minimisant la distance entre les deux poses :

$$F_p = \|\mathbf{p}_{k+1} - \mathbf{p}_k\|^2 \quad (4.16)$$

où \mathbf{p}_k est la position courante et \mathbf{p}_{k+1} la position désirée.

- Pour l'évitement des obstacles, la même fonction de coût définie dans la section 1 est considérée. Il est à rappeler que le but est de maximiser la distance entre la caméra et tous les obstacles en minimisant la fonction de coût définie par l'équation 3.53.
- Les deux fonctions de coût précédentes sont combinées linéairement :

$$F_c = \lambda_1 F_p + \lambda_2 O \quad (4.17)$$

où $\lambda_1, \lambda_2 > 0$ sont deux paramètres permettant de combiner les deux fonctions. Une fois de plus, la méthode de la descente du gradient va être adoptée pour minimiser F_c .

Dans cette approche, le contenu visuel pertinent est assuré par la maximisation de la fonction de coût mélange de gaussiennes basées saillance décrite en détail dans le chapitre 3. Les d.d.l disponibles de la caméra pour la pertinence sont les deux rotations restantes (la troisième rotation (le roulis) n'est pas considérée pour éviter les rotations irréalistes autour de l'axe optique de la caméra).

Il est à rappeler que la maximisation de la fonction de coût du mélange de gaussiennes basées saillance $G(\mathbf{r})$ est équivalente à la minimisation de son opposé $-G(\mathbf{r})$. Ainsi, la recherche du maximum revient à annuler la dérivée de cette fonction. Pour le calcul de la loi de contrôle du mélange de gaussiennes basées saillance, il est nécessaire de calculer la Jacobienne $d_{\mathbf{r}}(-G(\mathbf{r}))$ et le Hessien $d_{\mathbf{r}}^2(-G(\mathbf{r}))$. Leurs expressions ont été données dans le Chapitre 3, respectivement, par les équations 3.46 et 3.49.

b- Combinaison des contraintes :

En considérant le formalisme de redondance, les translations sont attribués à la loi de contrôle hybride (**Contrainte principale** et **Contrainte 1**), tandis que les rotations sont contrôlées par le mélange de gaussiennes basées saillance (**Contrainte 2**). La fonction globale qui prend en compte la tâche principale et les contraintes est donnée par :

$$\mathbf{e} = \mathbf{W}^+ \mathbf{W} F_c + (\mathbf{I}_6 - \mathbf{W}^+ \mathbf{W}) d_{\mathbf{r}}(-G(\mathbf{r}_k)) \quad (4.18)$$

où $\mathbf{W}^+ \mathbf{W}$ et $(\mathbf{I}_6 - \mathbf{W}^+ \mathbf{W})$ sont les deux opérateurs de projection, assurant que la tâche secondaire (**Contrainte 2**) ne va pas gêner la tâche principale (**Contrainte principale** et **Contrainte 1**).

Ainsi, la loi de contrôle basée redondance [Courty 2001] est donnée par la forme suivante :

$$\dot{\mathbf{r}}_k = -\lambda \dot{\mathbf{e}} + (\mathbf{I}_6 - \mathbf{W}^+ \mathbf{W}) \frac{\partial d_{\mathbf{r}}(-G(\mathbf{r}_k))}{\partial t} \quad (4.19)$$

4.2.2.2 Contenu similaire

En plus des contraintes définies ci-dessus, il est important de gérer le contenu de l'image au cours du déplacement de la caméra. Ceci veut dire que la caméra ne doit pas être orientée vers la même zone pertinente pendant un temps défini (éviter d'avoir un contenu similaire). Pour résoudre ce problème, deux solutions sont considérées. La première est un mécanisme permettant de changer automatiquement l'orientation de la caméra. La deuxième consiste à augmenter l'interaction de l'utilisateur en lui donnant la liberté d'intervenir dans le choix de l'orientation de la caméra s'il s'avère que le contenu de l'image est devenu quasiment statique.

Par exemple, si la scène considérée est simple avec deux objets intéressants à explorer, avec une disposition face à face de ces derniers comme illustré dans la figure 4.14. Alors à partir du point de vue initial de la caméra représenté en noir, l'objectif est d'explorer le premier objet positionné sur le côté gauche et par la suite de se diriger vers le second objet.

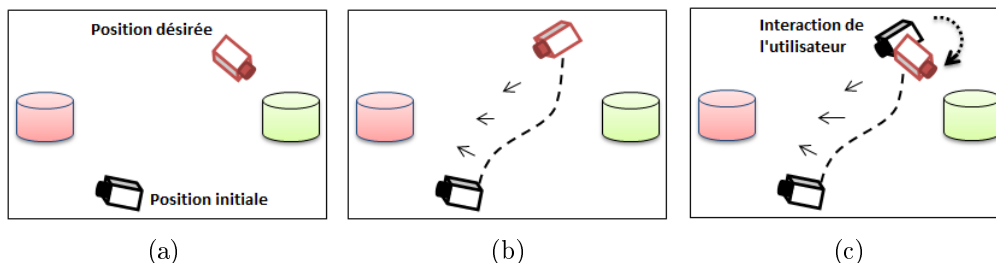


FIGURE 4.14 – (a) Les deux cylindres représentent les objets à inspecter. La position initiale de la caméra est en noir tandis que la position finale est en rouge, (b) La trajectoire finale de la caméra si son orientation est contrôlée par la primitive photométrique, les flèches définissent l'orientation de la caméra (un seul objet a été inspecté, celui à gauche). (c) Quand la position finale est atteinte, l'utilisateur change l'orientation de la caméra afin d'inspecter le second objet.

Cependant, en considérant la configuration de la scène illustrée par la figure 4.14, on remarque que la position initiale de la caméra virtuelle est orientée vers le cylindre rouge et que le cylindre vert n'est pas inclus dans son champ de vue. Il est clair que le contrôle de l'orientation de la caméra en maximisant la pertinence à l'aide de la primitive mélange de gaussiennes basées saillance n'est pas suffisant parce que la

caméra va se déplacer vers la position finale en ne gardant dans son champ de vue que le cylindre rouge. Par conséquent, lorsque l'information pertinente est présente dans deux côtés complètement opposés (dans cet exemple les deux cylindres) et que l'orientation initiale de la caméra pointe vers un seul objet (cylindre rouge), l'intervention de l'utilisateur s'avère nécessaire. En effet, en premier la caméra virtuelle se déplace vers la position désirée en contrôlant son orientation en maximisant la fonction de coût du mélange de gaussiennes basées saillance, ensuite l'utilisateur a la possibilité de modifier manuellement l'orientation de la caméra afin d'inspecter le second objet.

Une autre solution est proposée si l'utilisateur ne réagit pas. Il s'agit d'une méthode de changement automatique de l'orientation de la caméra.

Pour illustrer le principe de cette méthode, on considère qu'à chaque itération k , la loi de contrôle basée redondance génère une nouvelle pose \mathbf{p}_k . À chaque pose de la caméra virtuelle, on suppose une sphère centrée sur le point central perçu par la caméra \mathbf{m}_k avec un rayon défini r_k qui peut avoir la taille du champ de vision de la caméra. Chaque point central \mathbf{m}_k est relatif à une pose \mathbf{p}_k et un rayon r_k . Après un nombre N_p de poses générées, on calcule la distance moyenne d_k entre la position du point central \mathbf{m}_k et de $\frac{N_p}{2}$ points centraux des poses antérieures :

$$d_k = \frac{2}{N_p} \sum_{0 < j \leq \frac{N_p}{2}} \|\mathbf{m}_k - \mathbf{m}_j\| \quad (4.20)$$

Si la distance moyenne d_k est inférieure au rayon r_k de la sphère, cela signifie que durant les N_p itérations, la caméra a été orientée vers la même zone de l'environnement. En d'autres termes, le volume d'intersection des sphères est grand. Le système va alors déclencher un processus pour changer automatiquement l'orientation de la caméra soit en faisant un mouvement de panorama soit d'inclinaison.

4.2.2.3 Contrôle du flot optique

Pour prendre en compte la contrainte basée sur le calcul du flot optique permettant de régulariser le mouvement dans l'espace de l'image, un schéma hiérarchique en deux étapes est proposé.

La première étape consiste à calculer la loi de contrôle basée redondance décrite ci-dessus (Équation (4.19)). La deuxième étape lisse le mouvement. Pour ce faire, le flot optique $\hat{\mathbf{u}}$ entre l'image de synthèse courante et l'image rendue après l'application de la loi de contrôle basée redondance est calculé de la même façon que dans la section 4.1.2.4. La prochaine étape consiste à calculer la moyenne de ce flot

4.2. Exploration assistée : mélanges de gaussiennes basées saillance 111

optique $m_{\mathbf{u}}$ et à fixer une valeur moyenne désirée du flot optique entre les images courante et désirée à $m_{\mathbf{u}_{des}}$.

Ensuite, l'incrément de pose \mathbf{r}_k (Équation 3.34) est pondéré par un facteur calculé à partir du rapport entre la valeur moyenne désirée du flot et la moyenne calculée du flot. Cette pondération a pour but de réguler l'amplitude moyenne du mouvement.

Ainsi, la loi de contrôle finale hiérarchique est donnée par :

$$\dot{\mathbf{r}}_k = \frac{m_{\mathbf{u}_{des}}}{m_{\mathbf{u}}} \dot{\mathbf{r}}_k \quad (4.21)$$

Cette dernière loi de contrôle va permettre de prendre en compte toutes les contraintes présentées précédemment tout en lissant le mouvement dans l'image.

4.2.3 Résultats expérimentaux

L'utilisateur commence par définir un ensemble de positions 3D cibles dans le modèle 3D. A travers ces positions, il spécifie les zones à visiter. Ainsi, il décide de la trajectoire globale de la caméra virtuelle en la contraignant à passer à travers les positions définies. Le système prend donc en entrée les positions souhaitées dans le modèle 3D puis il génère une trajectoire de la caméra automatiquement en respectant les contraintes définies précédemment. Au fur et à mesure que la caméra se déplace dans le modèle, l'information pertinente est mise en avant à l'aide de la primitive image, qu'est le mélange de gaussiennes basées saillance qui agit sur les rotations de la caméra.

4.2.3.1 Résultats de simulation

Dans ce premier résultat, un environnement simple constitué de différents objets géométriques : cubes, cylindres et cônes est considéré. L'utilisateur définit un ensemble de points de passages P_k ($k = 1...4$) représentés à l'aide de pyramides jaunes et vertes (Figure 4.15). La caméra virtuelle se déplace de P_1 à P_4 en contrôlant son orientation à l'aide du mélange de gaussiennes basées saillance (**Contrainte 2**), en évitant les obstacles (**Contrainte 1**), en gardant une orientation pertinente (**Contrainte 3**) et enfin en assurant un contenu visuel qui se renouvelle (**Contrainte 5**).

La caméra se déplace entre les points de passage P_1 à P_2 en inspectant les trois objets figurant sur le côté A. Entre les poses P_2 et P_3 , l'algorithme détecte un contenu visuel similaire car la caméra garde dans son champ de vue les objets saillants du côté A. Elle effectue alors un changement automatique d'orientation dans le but d'explorer les objets géométriques figurant sur les cotés B et C. Ce

changement d'orientation est une rotation panoramique qui est représentée dans la figure 4.15 à l'aide d'une pyramide verte après P_2 . Ensuite, le système poursuit sa navigation afin de gagner les autres positions définies par l'utilisateur, notamment P_3 et P_4 .

L'algorithme permet de générer en ligne la trajectoire de la caméra qui est représentée en rouge dans la vue de dessus du modèle (Figure 4.15).

La figure 4.16 montre un extrait de la séquence d'images rendues (8 images). Tous les objets géométriques qui se trouvent dans le modèle et qui représentent l'information de pertinence, c'est-à-dire ce qui est intéressant à visiter par l'utilisateur, sont présents dans l'échantillon de la séquence d'images. Ainsi, le but de cette expérience, qui est d'explorer tout l'environnement 3D, a été atteint avec succès.

Afin de démontrer l'utilité du mélange de gaussiennes basées saillance sur le

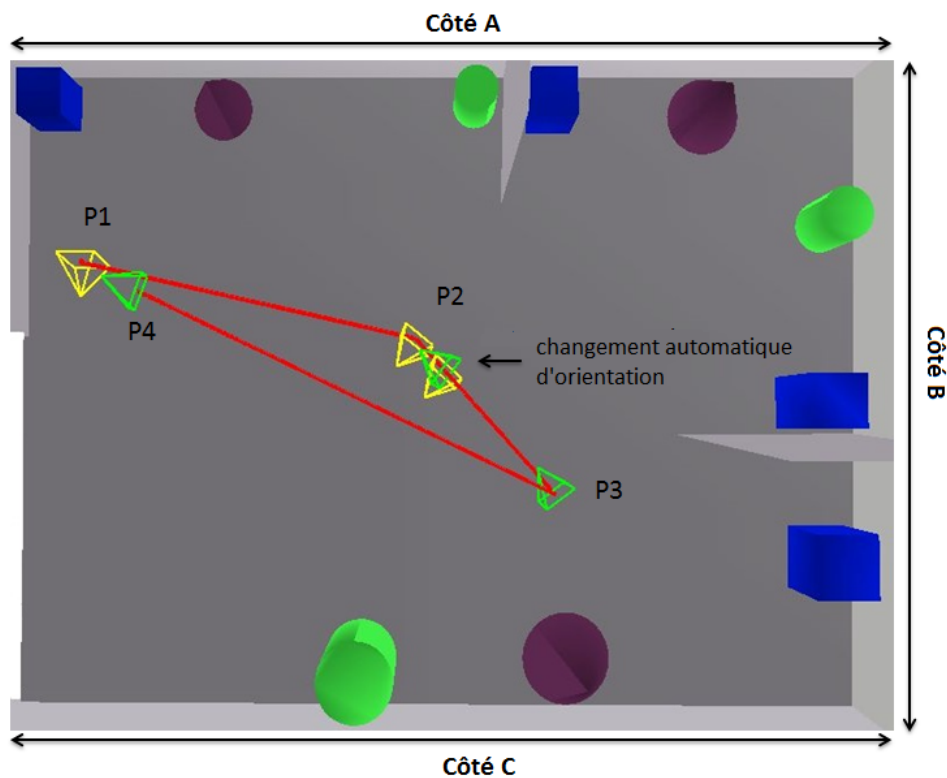


FIGURE 4.15 – Vue de dessus de l'environnement 3D : quatre cubes bleus, trois cylindres verts et trois cônes pourpres. Les positions cibles définies par l'utilisateur sont représentées par des pyramides jaunes et vertes et la trajectoire de la caméra avec une ligne rouge. Le système change automatiquement l'orientation de la caméra après la position P_2 .

4.2. Exploration assistée : mélanges de gaussiennes basées saillance 113

contrôle de l'orientation de la caméra virtuelle, une comparaison du comportement de la caméra entre les positions P_1 et P_2 en utilisant deux lois de commande a été réalisée. La première loi est calculée simplement à l'aide de l'asservissement visuel virtuel basé poses sur tous les d.d.l de la caméra. Alors que la deuxième loi de commande concerne le schéma de redondance en contrôlant l'orientation de la caméra à l'aide du mélange de gaussiennes basées saillance. En partant de la même pose initiale (la première image de synthèse est la même pour la figure 4.17a et la figure 4.17b), les objets géométriques sont observés à partir d'un mauvais point de vue comme illustré sur la séquence d'images de la figure 4.17b. Les objets sont tronqués dans l'image, contrairement à la loi de contrôle proposée où les objets sont bien visibles sur la séquence d'image illustrée par la figure 4.17a.

4.2.3.2 Résultats dans un environnement vaste et complexe

Cette approche est testée sur le modèle 3D de la cathédrale Notre Dame d'Amiens. En plus du modèle 3D, l'algorithme prend en entrée un ensemble de positions cibles définies par l'utilisateur P_k ($k = 1..5$). La trajectoire de la caméra obtenue respecte toutes les contraintes posées par le système (Algorithme 1).

Partant de la position de départ P_1 , où la caméra est orientée vers les voûtes et les vitraux (Figure 4.18), la caméra virtuelle se déplace dans l'environnement 3D en passant par toutes les positions cibles définies par l'utilisateur en utilisant la loi de commande hiérarchique calculée dans la section précédente. La caméra évite

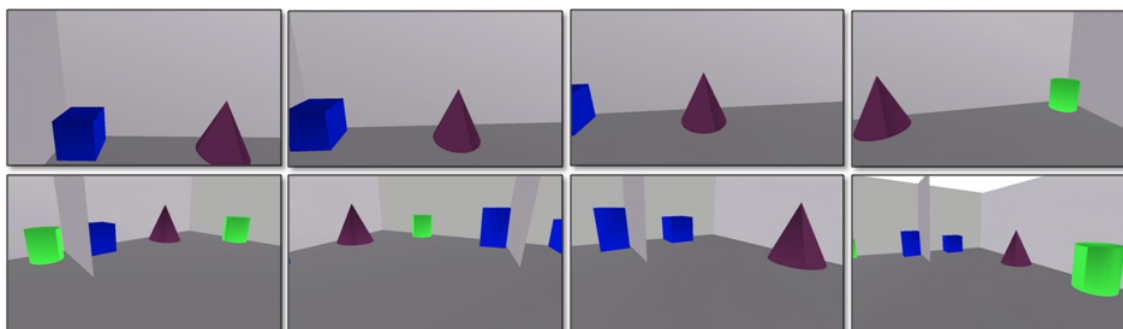


FIGURE 4.16 – Séquence d'images du résultat de la navigation. Tous les objets géométriques de la figure 4.15 sont présents dans la séquence d'images. De la première à la quatrième image, les objets du côté A sont observés. Dans la cinquième et la sixième image, tous les objets du côté B sont observés, tandis que dans la septième et la huitième image tous les objets du côté C sont présents.

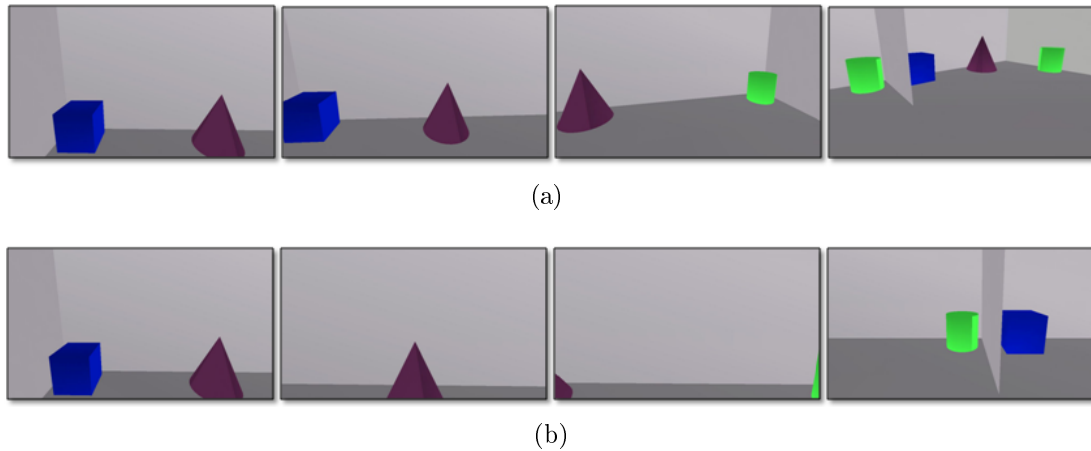


FIGURE 4.17 – (a) Un échantillon des images rendues en utilisant la primitive photométrique pour le contrôle de l'orientation, (b) Un échantillon des images rendues en utilisant l'asservissement visuel basé poses.

les obstacles tout en gardant un mouvement régulier dans l'image à l'aide de la contrainte basée sur le calcul du flot optique. Ceci est plus visible sur la vidéo rendue.

La trajectoire suivie par la caméra en passant par les positions définies par l'utilisateur est représenté dans la figure 4.19.

Pour illustrer le fonctionnement de la contrainte d'évitement d'obstacles, deux positions de la caméra sont choisies dans l'environnement P_1 et P_2 . Pour la première position, l'orientation de la caméra est définie, ce qui n'est pas le cas pour la seconde. En effet, l'orientation de la caméra durant son déplacement de P_1 à P_2 est contrôlée par le mélange de gaussiennes basées saillance. Pour la figure 4.20, l'algorithme hiérarchique est appliqué afin de se déplacer d'une position à l'autre, de telle façon à respecter toutes les contraintes. On peut remarquer que la caméra réussit à éviter les obstacles. Alors, que dans la figure 4.21 la contrainte de l'évitement des obstacles n'est pas prise en compte, laissant la caméra à traverser le mur. Sur ces deux figures, seule une partie de l'environnement a été représentée afin de bien visualiser la trajectoire empruntée par la caméra. Cependant, l'image rendue à partir de la position P_2 est illustrée pour avoir une idée sur le point de vue finale de la caméra.

4.2.4 Conclusion

Le système d'aide à la navigation que nous avons proposé repose sur l'utilisation d'un algorithme hiérarchique basé sur le formalisme de redondance. Cette technique

4.2. Exploration assistée : mélanges de gaussiennes basées saillance 115

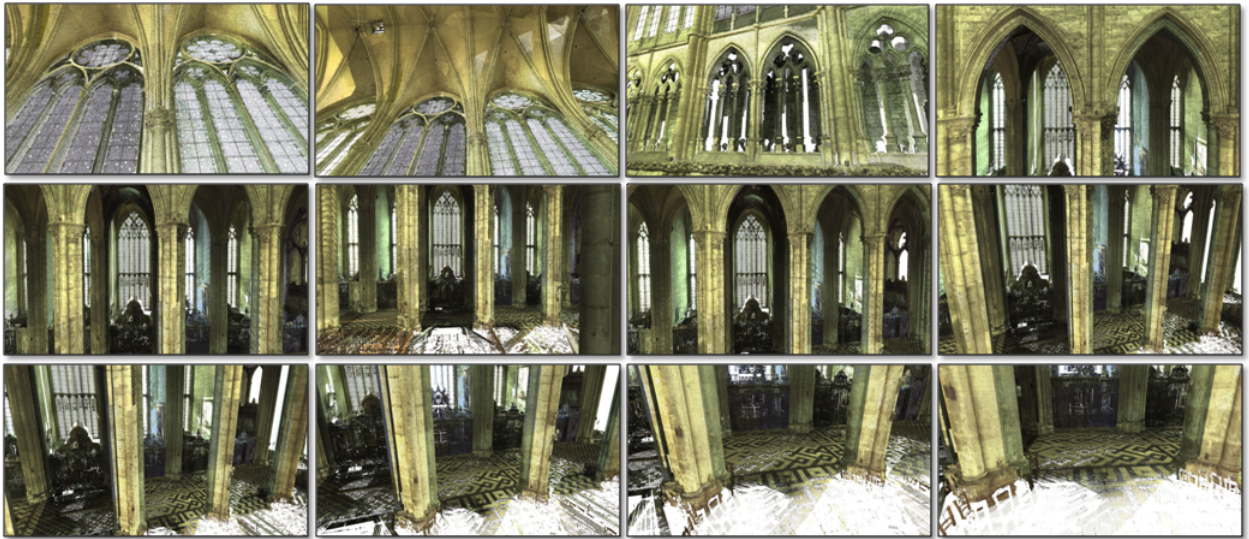


FIGURE 4.18 – Séquence d'images pour la navigation assistée.

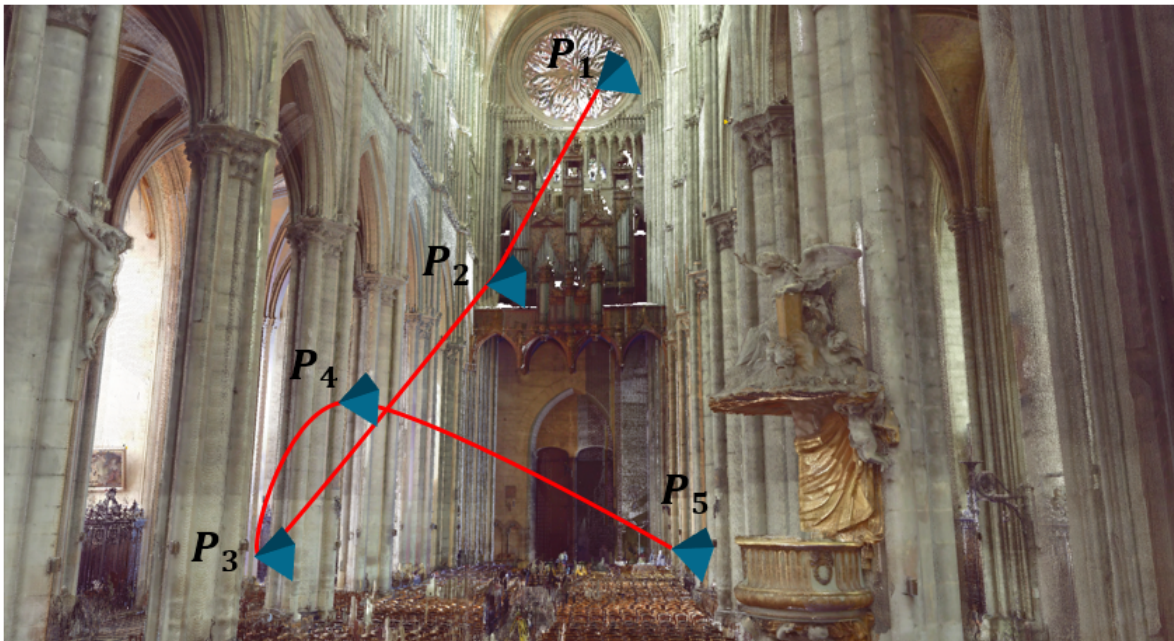


FIGURE 4.19 – Les positions cibles sont en bleu et la trajectoire de la caméra en rouge.

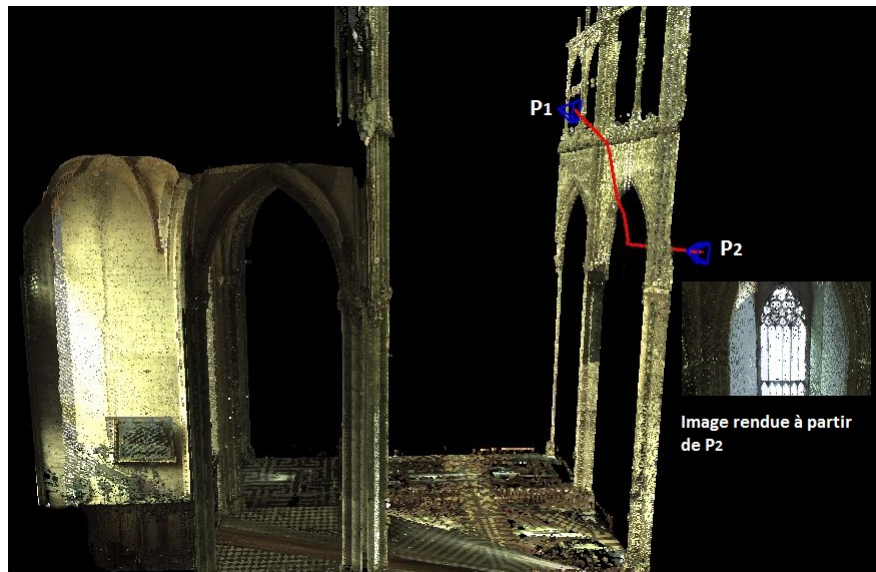


FIGURE 4.20 – Trajectoire de la caméra en respectant la contrainte de l'évitement des obstacles.

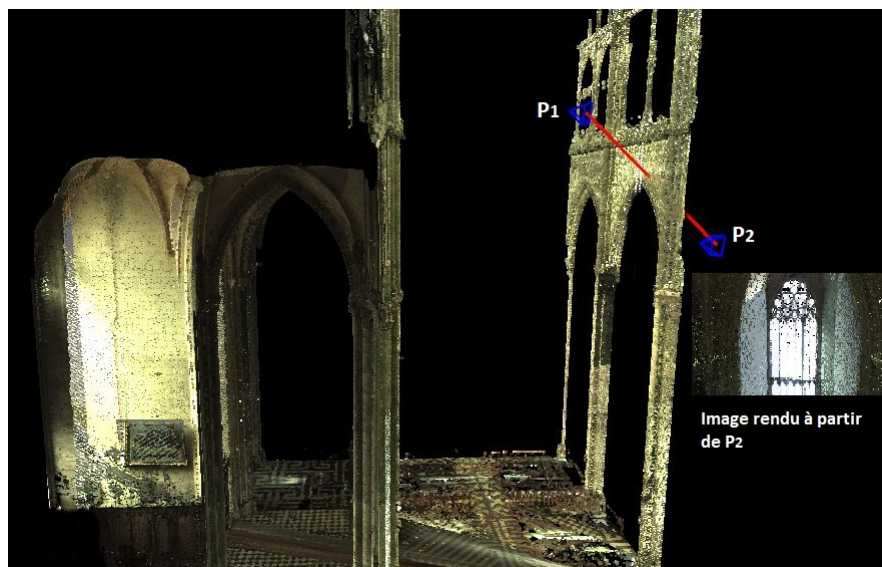


FIGURE 4.21 – Trajectoire de la caméra sans respecter la contrainte de l'évitement des obstacles.

4.2. Exploration assistée : mélanges de gaussiennes basées saillance 117

permet de prendre en compte toutes les contraintes nécessaires définies, qui servent à la construction d'une trajectoire de la caméra assurant une exploration pertinente et réaliste. L'approche a été testée dans un environnement synthétique simple pour une première validation. Ensuite, elle a été utilisée pour l'exploration d'un nuage 3D plus vaste et complexe, celui de la cathédrale Notre dame d'Amiens. Ce système d'aide à la navigation virtuelle fournit à l'utilisateur la possibilité de visiter les zones qui l'intéressent dans un environnement quelconque, tout en lui offrant un résultat visuel pertinent et réaliste. Les contraintes définies semblent suffisantes pour ce type d'application. En outre, cette approche est générale et peut donc intégrer d'autres contraintes sous la forme de critères à optimiser.

Conclusion et perspectives

L'objectif du projet dans lequel s'inscrit la thèse, est de développer un système d'aide à la navigation dans la maquette 3D de la cathédrale d'Amiens. Problème difficile parce qu'il faut automatiser tout en respectant ce qui dépend de la volonté de l'utilisateur. Néanmoins, dans ce manuscrit, la tâche de navigation a été formalisée à l'aide de critères qui exploitent l'information 2D et 3D du modèle. Nous avons tout d'abord développé le pilotage automatique, puis introduit l'interaction. Pour cela, nous avons utilisé le formalisme de l'asservissement visuel virtuel pour le contrôle des d.d.l de la caméra.

Dans un premier temps, une navigation locale ou ce que l'on a appelé "le cadrage visuel" est proposé. Ce problème se base sur l'exploitation d'une primitive visuelle afin de pouvoir cadrer l'information pertinente dans un environnement. Deux primitives visuelles photométriques ont été étudiées pour la modélisation de la pertinence visuelle : l'entropie photométrique et la saillance visuelle. Cependant, leur utilisation directe pour le cadrage visuel n'est pas appropriée. L'entropie photométrique souffre des problèmes d'optimums locaux surtout quand les niveaux de gris dans l'histogramme utilisé pour son calcul n'est pas équilibré, ce qui influence négativement la convergence. En ce qui concerne l'utilisation de la saillance visuelle directement, la fonction de coût dédiée au contrôle de la caméra n'est autre que la somme des intensités de saillance. Le domaine de convergence est très restreint, dans la mesure où il faut être très proche de la solution pour converger. Pour résoudre ce problème, l'image est, pour la première fois, considérée d'une manière différente à l'aide d'un mélange de gaussiennes basées saillance. Le point fort de cette nouvelle primitive est que les intensités de saillance de chaque pixel sont représentées par une fonction continue dont le domaine de définition est $]-\infty, \infty[$. Cette primitive a démontré son efficacité pour la résolution du problème de cadrage visuel. Elle permet d'améliorer nettement la convergence et la robustesse du cadrage visuel. Différentes expérimentations en simulation, sur un nuage de points 3D et en réel ont été réalisées pour la validation de l'approche proposée.

Cette méthode de cadrage automatique a été étendue à une méthode assistée. Nous avons combiné le mélange de gaussiennes basées saillance d'une manière adéquate avec l'interaction de l'utilisateur. Cette interaction agit sur la translation selon l'axe \vec{Z} de la caméra, afin de permettre à l'utilisateur de s'approcher ou de s'éloigner de l'objet considéré pour le cadrage, en temps réel, grâce à l'utilisation

d'images de basse résolution, sans gêner son comportement. Une étude comparative basée sur l'avis d'un échantillon d'utilisateurs a été réalisée pour la validation qualitative de cette méthode de cadrage visuel assisté.

Ensuite, dans le quatrième chapitre, nous avons présenté deux méthodes dédiées à l'exploration ou à la navigation globale dans un gigantesque et complexe environnement virtuel constitué de nuage de points 3D colorés.

La première méthode permet de réaliser une exploration automatique avec pour entrées du système une pose initiale de la caméra (position et orientation) en plus du modèle 3D. L'approche est formalisée sous la forme d'une tâche principale et de contraintes. La tâche principale, relative à la pertinence visuelle, définit la quantité de l'information perçue dans une image acquise à partir d'une pose de la caméra. Elle a été modélisée en se basant sur la théorie de l'information et plus précisément sur l'entropie de Shannon. L'utilisation de l'entropie photométrique dans ce type d'environnement, avec une distribution photométrique très riche diminue le problème des optimums locaux. Néanmoins, pour régler définitivement ce problème, nous avons mis en œuvre un mécanisme qui permet d'alterner entre la maximisation et la minimisation de l'entropie photométrique. Ensuite, nous avons introduit différentes contraintes pour garantir un mouvement réaliste en gérant l'évitement des obstacles, la fluidité du mouvement et la cohérence des rotations de la caméra. Le schéma de résolution du système se fait d'une manière hiérarchique. La direction du mouvement est d'abord calculée en combinant à l'aide d'une loi de contrôle hybride basée sur la maximisation (ou la minimisation) de l'entropie photométrique et l'évitement des obstacles. Par la suite, l'amplitude est réglée en prenant en compte la contrainte basée sur le calcul du flot optique. La loi de contrôle finale se restreint à cinq d.d.l. L'approche a été testée sur la maquette virtuelle de la cathédrale.

Finalement, nous avons proposé un système d'aide à la navigation virtuelle. L'utilisateur définit des positions dans le modèle 3D reflétant les zones qu'il souhaite visiter. Le problème est formalisé sous la forme de plusieurs contraintes nécessaires à la réalisation de la tâche de navigation. On commence par combiner, à l'aide d'une loi de commande hybride, l'asservissement visuel basé pose, qui permet le déplacement de la caméra à travers des poses définies par l'utilisateur, avec l'évitement des obstacles. Ensuite, on introduit le formalisme de redondance afin de prendre en compte la contrainte modélisée par le mélange de gaussiennes basées saillance qui assure une orientation pertinente de la caméra. Enfin, la loi de contrôle basée redondance est combinée hiérarchiquement avec une contrainte de fluidité du mouvement et une contrainte de non-similarité du contenu de l'image. Cette approche a été testée en simulation et aussi sur l'environnement de la cathédrale Notre Dame

d'Amiens.

L'approche proposée pour réaliser le cadrage visuel fournit des résultats intéressants et précis pour différents objets géométriquement différents. La convergence est assurée dans tous les cas et le temps de calcul est restreint en raison de l'utilisation d'images de très basse résolution et d'un environnement de taille raisonnable. Cependant, quand le but est d'explorer un environnement très volumineux, le temps de calcul constitue un véritable frein. Il représente le principal point faible des méthodes d'explorations abordées dans ce mémoire. Ainsi l'interaction directe de l'utilisateur n'est pas encore possible pour l'instant, sauf à considérer une sous partie d'un environnement complexe.

Plusieurs perspectives de recherche s'ouvrent à partir de cette thèse. Dans un premier temps, il serait utile et voire important d'étudier la stabilité de la méthode de cadrage visuel qui est résolue à l'aide du formalisme de l'asservissement visuel. Bien que le mélange de gaussiennes basées saillance permet de converger à un résultat très concluant, il n'a été évalué que d'une manière purement visuelle. L'étude de stabilité pourrait affirmer si la position finale atteinte par la caméra à convergence, reflète exactement la solution globale du problème et qu'il ne s'agit pas juste d'une solution locale.

Toujours dans le cas du cadrage visuel, la méthode proposée pourrait être étendue à l'exploration et la compréhension d'un environnement plus complexe. Prenant l'exemple d'une ou de plusieurs pièces contenant différents objets, l'application consistera à la découverte de cette environnement inconnu en utilisant un robot mobile doté d'une camera orientable et inclinable.

Dans ce travail, la méthode de résolution, des différentes approches de contrôle de caméra, appliquée au cadrage visuel et à l'exploration repose sur les principes de l'optimisation non linéaire. D'autres critères peuvent être utilisés pour la description de la pertinence visuelle. Au lieu de n'utiliser que des primitives photométriques, il serait intéressant d'explorer quelques primitives géométriques, notamment la profondeur ou les droites. Ces dernières pourrait faire preuve d'une piste intéressante surtout dans le cas de l'exploration de la maquette de la cathédrale qui est constituée de plusieurs piliers sous la forme de colonnes.

Diminuer les temps de calcul est la perspective la plus importante pour rendre tout interactif dans ce travail. La méthode de rendu utilisée est très lourde. En effet, pour obtenir une image virtuelle à partir d'une pose de la caméra, il faut parcourir l'ensemble des sommets du modèle et vérifier pour chaque sommet s'il est dans le cône de visibilité. Ensuite, une projection 3D est effectuée pour l'ensemble des points appartenant à ce cône de visibilité. Ce processus requiert un temps de

calcul très élevé surtout quand il s'agit d'un modèle avec une énorme quantité de données comme celui de la cathédrale (2 milliards de points). Une solution serait de structurer le modèle afin d'organiser les données et de connaître exactement les points qui sont visibles à partir d'une pose de la caméra. Il y a aussi la possibilité de gérer le niveau de détails, entre autre de ne charger que les parties du modèle qui sont visibles à partir d'une pose de la caméra. Si, par exemple, à partir d'une certaine pose, on a une vue globale de l'environnement. Il serait plus adéquat de ne pas charger l'intégralité des points 3D, mais seulement ceux qui sont utiles à la visualisation de la scène. Ensuite, au fur et à mesure que la caméra évolue dans le modèle pour se rapprocher d'une zone bien précise, les points 3D relatifs à cette dernière seront chargés pour la visualiser plus en détails.

Une fois le problème du temps de calcul résolu, une évaluation qualitative du système de navigation assisté serait intéressante par un échantillon d'utilisateurs. Cette évaluation pourrait être sous la forme d'un questionnaire qui traite à la fois de la pertinence visuelle qui est plus au moins subjective, de la qualité et la cohérence des mouvements de la caméra virtuelle.

Bibliographie

- [Andújar 2004] C. Andújar, P. Vázquez et M. Fairén. *Way-Finder : guided tours through complex walkthrough models*. In Computer Graphics Forum, volume 23, pages 499–508, 2004. (Cité en pages [iv](#), [39](#), [40](#) et [103](#).)
- [Astley 1994] Mark Astley et Mitchell Roth. *Spline animator : Smooth camera motion for avs animation*. In Proceedings of the 1994 International AVS Users Conference, pages 142–151. Citeseer, 1994. (Cité en page [38](#).)
- [Bakthavatchalam 2013] M. Bakthavatchalam, F. Chaumette et E. Marchand. *Photometric moments : New promising candidates for visual servoing*. In IEEE International Conference on Robotics and Automation (ICRA),, pages 5241–5246, 2013. (Cité en page [18](#).)
- [Bandi 1998] S. Bandi et D. Thalmann. *Space discretization for efficient human navigation*. In Computer Graphics Forum, volume 17, pages 195–206, 1998. (Cité en page [39](#).)
- [Bares 1998] W. Bares, L. Zettlemoyer, D. Rodriguez et J. Lester. *Task-sensitive cinematography interfaces for interactive 3D learning environments*. In Proceedings of the 3rd international conference on Intelligent user interfaces ACM, pages 81–88, 1998. (Cité en page [90](#).)
- [Barral 2000] P. Barral, G. Dorme et D. Plemenos. *Scene understanding techniques using a virtual camera*. In Proceedings of Eurographics, volume 19, 2000. (Cité en pages [26](#) et [29](#).)
- [Beckhaus 2000] S. Beckhaus, F. Ritter et T. Strothotte. *Cubicalpath-dynamic potential fields for guided exploration in virtual environments*. In The Eighth Pacific Conference on Computer Graphics and Application IEEE., pages 387–459, 2000. (Cité en page [41](#).)
- [Berger 1997] M-O. Berger. *Resolving Occlusion in Augmented Reality : a Contour Based Approach without 3D Reconstruction*. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pages 91–96, 1997. (Cité en page [18](#).)
- [Bhattacharya 2006] S. Bhattacharya, N. Atay, G. Alankus, C. Lu, O. Bayazit et G. Roman. *Roadmap query for sensor network assisted navigation in dynamic environments*. In Distributed Computing in Sensor Systems, pages 17–36. 2006. (Cité en page [39](#).)

- [Björnsson 2005] Y. Björnsson, M. Enzenberger, R. Holte et J. Schaeffer. *Fringe Search : Beating A* at Pathfinding on Game Maps*. CIG, vol. 5, pages 125–132, 2005. (Cité en page 38.)
- [Bourne 2008] O. Bourne, A. Sattar et S. Goodwin. *A constraint-based autonomous 3d camera system*. Constraints, vol. 13, pages 180–205, 2008. (Cité en pages 90 et 104.)
- [Bruce 2005] Neil Bruce et John Tsotsos. *Saliency based on information maximization*. In Advances in neural information processing systems, pages 155–162, 2005. (Cité en pages 27 et 28.)
- [Canny 1988] J. Canny. The complexity of robot motion planning. MIT press, 1988. (Cité en page 39.)
- [Caron 2011] G. Caron et D. Eynard. *Multiple camera types simultaneous stereo calibration*. In IEEE International Conference on Robotics and Automation (ICRA), pages 2933–2938. IEEE, 2011. (Cité en page 11.)
- [Chaumette 1990] F. Chaumette. *La relation vision-commande : théorie et application à des tâches robotiques*. PhD thesis, 1990. (Cité en pages 14 et 15.)
- [Chaumette 2004] F. Chaumette. *Image moments : a general and useful set of features for visual servoing*. IEEE Transactions on Robotics, vol. 20, no. 4, pages 713–723, 2004. (Cité en pages 15 et 18.)
- [Chaumette 2006] F. Chaumette et S. Hutchinson. *Visual servo control. I. Basic approaches*. Robotics & Automation Magazine, IEEE, vol. 13, pages 82–90, 2006. (Cité en pages 12 et 18.)
- [Chen 1988] M. Chen, S. Mountford et A. Sellen. *A study in interactive 3-D rotation using 2-D control devices*. In ACM SIGGRAPH Computer Graphics, volume 22, pages 121–129, 1988. (Cité en page 35.)
- [Chiaverini 1997] S. Chiaverini. *Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators*. In IEEE Transactions on Robotics and Automation, volume 13, pages 398–410, 1997. (Cité en pages 21 et 104.)
- [Christie 2005] M. Christie, R. Machap, J-M. Normand, P. Olivier et J. Pickering. *Virtual camera planning : A survey*. In Smart Graphics, pages 40–52, 2005. (Cité en page 42.)
- [Collewet 2008] C. Collewet, E. Marchand et F. Chaumette. *Visual servoing set free from image processing*. In IEEE International Conference on Robotics and Automation (ICRA), pages 81–86. IEEE, 2008. (Cité en pages 16, 17 et 50.)

- [Courty 2001] N. Courty et E. Marchand. *Computer animation : A new application for image-based visual servoing*. In IEEE International Conference on Robotics and Automation (ICRA), volume 1, pages 223–228, 2001. (Cité en pages iv, 18, 19, 26, 27, 41, 68, 90, 104, 106 et 108.)
- [Courty 2002] N. Courty. *Animation référencée vision : de la tâche au comportement*. PhD thesis, PhD thesis, 2002. (Cité en pages iii, 20 et 21.)
- [Crombez 2014] N. Crombez, G. Caron et E. Mouaddib. *Colorisation de nuages de points 3D par recalage dense d'images numériques*. TS. Traitement du signal, vol. 31, no. 1-2, pages 81–106, 2014. (Cité en page 79.)
- [Crombez 2015] N. Crombez, G. Caron et E. Mouaddib. *Photometric Gaussians mixture based visual servoing*. In IEEE/RSJ International Conference on Intelligent Robots and Systems,(IROS), 2015. (Cité en page 18.)
- [Dame 2009] A. Dame et E. Marchand. *Entropy-based visual servoing*. In IEEE International Conference on Robotics and Automation (ICRA), pages 707–713, 2009. (Cité en page 47.)
- [Dame 2011] A. Dame et E. Marchand. *Mutual information-based visual servoing*. IEEE Transactions on Robotics, vol. 27, pages 958–969, 2011. (Cité en page 18.)
- [Deguchi 2000] K. 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 17.)
- [Dorme 2001] G. Dorme. *Study and implementation of 3D scenes comprehension techniques*. PhD thesis, PhD thesis, 2001. (Cité en page 29.)
- [Duane 1971] B. Duane. *Close-range camera calibration*. Photogram. Eng. Remote Sens, vol. 37, pages 855–866, 1971. (Cité en page 11.)
- [Espiau 1992] B. Espiau, f. Chaumette et P. Rives. *A new approach to visual servoing in robotics*. IEEE Transactions on Robotics and Automation, vol. 8, no. 3, pages 313–326, 1992. (Cité en pages 14 et 15.)
- [Feddema 1989] J. Feddema, T. Mitchell et R. Owen. *Vision-guided servoing with feature-based trajectory generation for robots*. In IEEE Transactions on Robotics and Automation, volume 5, pages 691–700, 1989. (Cité en pages 14, 15 et 90.)
- [Gleicher 1992] M. Gleicher et A. Witkin. *Through-the-lens camera control*. In ACM SIGGRAPH Computer Graphics, volume 26, pages 331–340, 1992. (Cité en page 41.)

- [Gumhold 2002] S. Gumhold. *Maximum entropy light source placement*. In IEEE Visualization, pages 275–282, 2002. (Cité en page 30.)
- [Hachet 2008] M. Hachet, F. Declé et S. Knödel P. Guitton. *Navidget for easy 3d camera positioning from 2d inputs*. In IEEE Symposium on 3D User Interfaces (3DUI), pages 83–89. IEEE, 2008. (Cité en pages iii, 37 et 38.)
- [Hafez 2008] A. Hafez, A. Supreeth et CV Jawahar. *Visual servoing based on gaussian mixture models*. In IEEE International Conference on Robotics and Automation (ICRA), pages 3225–3230, 2008. (Cité en page 15.)
- [Hagedorn 2008] B. Hagedorn et J. Döllner. *Sketch-based navigation in 3d virtual environments*. In Smart Graphics, pages 239–246, 2008. (Cité en pages iii, 38 et 39.)
- [Hager 1997] G-D. Hager. *A modular system for robust positioning using feedback from stereo vision*. In IEEE Transactions on Robotics and Automation, volume 13, pages 582–595, 1997. (Cité en pages 71 et 90.)
- [Harel 2006] Jonathan Harel, Christof Koch et Pietro Perona. *Graph-based visual saliency*. In Advances in neural information processing systems, pages 545–552, 2006. (Cité en pages 27 et 28.)
- [Hutchinson 1996] S. Hutchinson, G. Hager, P. Corke et al. *A tutorial on visual servo control*. IEEE Transactions on Robotics and Automation, vol. 12, no. 5, pages 651–670, 1996. (Cité en page 12.)
- [Igarashi 1998] T. Igarashi, R. Kadobayashi, K. Mase et H. Tanaka. *Path drawing for 3D walkthrough*. In Proceedings of the 11th annual ACM symposium on User interface software and technology, pages 173–174, 1998. (Cité en page 37.)
- [Itti 1998] L. Itti, C. Koch et E. Niebur. *A model of saliency-based visual attention for rapid scene analysis*. In IEEE Transactions on pattern analysis and machine intelligence, volume 20, pages 1254–1259, 1998. (Cité en pages 26, 27 et 55.)
- [Kallem 2007] V. Kallem, M. Dewan, J. Swensen, G. Hager et N. Cowan. *Kernel-based visual servoing*. In Intelligent Robots and Systems (IROS), pages 1975–1980. IEEE, 2007. (Cité en page 17.)
- [Kamada 1988] T. Kamada et S. Kawai. *A simple method for computing general position in displaying three-dimensional objects*. Computer Vision, Graphics, and Image Processing, vol. 41, no. 1, pages 43–56, 1988. (Cité en pages 26 et 29.)

- [Khan 2005] A. Khan, B. Komalo, J. Stam, G. Fitzmaurice et G. Kurtenbach. *Hovercam : interactive 3d navigation for proximal object inspection*. In Interactive 3D graphics and games ACM, pages 73–80, 2005. (Cité en pages iii et 37.)
- [Khatib 1986] O. Khatib. *Real-time obstacle avoidance for manipulators and mobile robots*. The international journal of robotics research, vol. 5, pages 90–98, 1986. (Cité en page 40.)
- [Lamarche 2009] F. Lamarche. *Topoplan : a topological path planner for real time human navigation under floor and ceiling constraints*. In Computer Graphics Forum, volume 28, pages 649–658, 2009. (Cité en page 39.)
- [Lapresté 2004] J. Lapresté et Y. Mezouar. *A Hessian approach to visual servoing*. In IEEE/RSJ International Conference on Intelligent Robots and Systems,(IROS), volume 1, pages 998–1003, 2004. (Cité en page 50.)
- [Lasseter 1987] J. Lasseter. *Principles of traditional animation applied to 3D computer animation*. In ACM Siggraph Computer Graphics, volume 21, pages 35–44, 1987. (Cité en page 104.)
- [Lee 2005] C H. Lee, A. Varshney et D W. Jacobs. *Mesh saliency*. In ACM Transactions on Graphics (TOG), volume 24, pages 659–666, 2005. (Cité en pages iii, 31 et 103.)
- [Li 1999] T. Li et T. Yu. *Planning tracking motions for an intelligent virtual camera*. In IEEE Robotics and Automation, volume 2, pages 1353–1358, 1999. (Cité en page 39.)
- [Mackinlay 1990] J. Mackinlay, S. Card et G. Robertson. *Rapid controlled movement through a virtual 3D workspace*. In ACM SIGGRAPH Computer Graphics, volume 24, pages 171–176, 1990. (Cité en page 37.)
- [Malis 1999] E. Malis, F. Chaumette et S. Boudet. *2-1/2D visual servoing*. Robotics & Automation Magazine, IEEE, vol. 15, pages 238–250, 1999. (Cité en page 19.)
- [Marchand 1998] E. Marchand et G. Hager. *Dynamic sensor planning in visual servoing*. In Proceedings International Conference on Robotics and Automation, volume 3, pages 1988–1993, 1998. (Cité en page 90.)
- [Marchand 2002a] É. Marchand et F. Chaumette. *Virtual Visual Servoing : a framework for real-time augmented reality*. In Computer Graphics Forum, volume 21, pages 289–297, 2002. (Cité en page 19.)

- [Marchand 2002b] É. Marchand et N. Courty. *Controlling a camera in a virtual environment*. The Visual Computer, vol. 18, no. 1, pages 1–19, 2002. (Cité en page 22.)
- [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*. Robotics & Automation Magazine, IEEE, vol. 12, pages 40–52, 2005. (Cité en page 73.)
- [Marchand 2007] E. Marchand. *Control camera and light source positions using image gradient information*. In IEEE International Conference on Robotics and Automation, pages 417–422, 2007. (Cité en pages iii, 22, 23 et 42.)
- [Martinet 1999] P. Martinet. *Comparison of visual servoing techniques : Experimental results*. In Proceedings of the European Control Conference, 1999. (Cité en page 18.)
- [Nayar 1996] S. Nayar, S. Nene, H. Murase et al. *Subspace methods for robot vision*. IEEE Transactions on Robotics and Automation, vol. 12, no. 5, pages 750–758, 1996. (Cité en page 17.)
- [Nelson 1994] B. Nelson et P. Khosla. *Integrating sensor placement and visual tracking strategies*. In Experimental Robotics III, pages 167–181. Springer, 1994. (Cité en page 22.)
- [Nilsson 1969] N. Nilsson. *A mobile automaton : An application of artificial intelligence techniques*. Rapport technique, DTIC Document, 1969. (Cité en page 39.)
- [Ogr] www.ogre3d.org. (Cité en pages 51, 72, 79 et 97.)
- [Osborne 1990] C. Ware S. Osborne. *Exploration and virtual camera control in virtual three dimensional environments*. In ACM SIGGRAPH Computer Graphics, volume 24, pages 175–183, 1990. (Cité en page 35.)
- [Paik 1998] D. Paik, C. Beaulieu, B. Jeffrey, G. Rubin et S. Napel. *Automated flight path planning for virtual endoscopy*. Medical Physics, vol. 25, no. 5, pages 629–637, 1998. (Cité en page 41.)
- [Palmer 1999] S. Palmer. *Vision science : Photons to phenomenology*, volume 1. MIT press Cambridge, MA, 1999. (Cité en page 27.)
- [Plemenos 1996] D. Plemenos et M. Benayada. *Intelligent display in scene modeling. new techniques to automatically compute good views*. In International Conference on Computer Graphics & Vision, 1996. (Cité en pages 26 et 29.)

- [Radhakrishna 2009] A. Radhakrishna, H. Sheila, E. Francisco et S. Sabine. *Frequency-tuned salient region detection*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1597–1604, 2009. (Cité en pages iii, 28, 29, 55 et 62.)
- [Rosen 1961] JB. Rosen. *The gradient projection method for nonlinear programming. Part II. Nonlinear constraints*. In Journal of the Society for Industrial & Applied Mathematics, volume 9, pages 514–532, 1961. (Cité en pages 21 et 104.)
- [Salomon 2003] B. Salomon, M. Garber, M. Lin et D. Manocha. *Interactive navigation in complex environments using path planning*. In Interactive 3D graphics, ACM, pages 41–50, 2003. (Cité en page 39.)
- [Shannon 1948] C. E. Shannon. *A mathematical theory of communication*. Bell system technical journal, vol. 27, 1948. (Cité en page 46.)
- [Sharir 1983] M. Sharir, C. Yapet *al.* *Retraction : A new approach to motion-planning*. In Proceedings of the fifteenth annual ACM symposium on Theory of computing, pages 207–220, 1983. (Cité en page 39.)
- [Shoemake 1992] K. Shoemake. *ARCBALL : a user interface for specifying three-dimensional orientation using a mouse*. In Graphics Interface, volume 92, pages 151–156, 1992. (Cité en page 36.)
- [Shtrom 2013] Elizabeth Shtrom, George Leifman et Ayellet Tal. *Saliency Detection in Large Point Sets*. In IEEE International Conference on Computer Vision (ICCV), pages 3591–3598, 2013. (Cité en pages iii, 27, 31, 32, 33, 34 et 103.)
- [Stuerzlinger 1999] W. Stuerzlinger. *Imaging all visible surfaces*. In Graphics Interface, volume 99, pages 115–122, 1999. (Cité en page 26.)
- [Sun 2005] Z. Sun, D. Hsu, T. Jiang, H. Kurniawati et J. Reif. *Narrow passage sampling for probabilistic roadmap planning*. IEEE Transactions on Robotics,, vol. 21, no. 6, pages 1105–1115, 2005. (Cité en page 39.)
- [T. Li 2008] Tsai-Yen T. Li et C. Cheng. *Real-time camera planning for navigation in virtual environments*. In Smart Graphics Springer, pages 118–129, 2008. (Cité en pages iv, 39 et 40.)
- [Tahri 2005] O. Tahri et F. Chaumette. *Point-based and region-based image moments for visual servoing of planar objects*. IEEE Transactions on Robotics,, vol. 21, no. 6, pages 1116–1127, 2005. (Cité en pages 15 et 18.)
- [Tan 2001] D. Tan, G. Robertson et Mary M. Czerwinski. *Exploring 3D navigation : combining speed-coupled flying with orbiting*. In Proceedings of the SIGCHI

- conference on Human factors in computing systems, pages 418–425, 2001. (Cité en page 37.)
- [Tomlinson 2000] B. Tomlinson, B. Blumberg et D. Nain. *Expressive autonomous cinematography for interactive virtual environments*. In Proceedings of the fourth international conference on Autonomous agents ACM, pages 317–324, 2000. (Cité en page 90.)
- [Treisman 1980] A. Treisman et G. Gelade. *A feature-integration theory of attention*. Cognitive psychology, Elsevier, vol. 12, pages 97–136, 1980. (Cité en page 27.)
- [Turner 1991] R. Turner, F. Balaguer et E. Gobbetti D. Thalmann. *Physically-based interactive camera motion control using 3D input devices*. In Scientific visualization of physical phenomena, pages 135–145. 1991. (Cité en page 37.)
- [van Waveren 2008] J.M.P. van Waveren et L.J.M. Rothkrantz. *Automated static and dynamic obstacle avoidance in arbitrary 3D polygonal worlds*. In In-Tech Education and Planning, 2008. (Cité en page 104.)
- [Vázquez 2001] P. Vázquez, M. Feixas, M. Sbert et W. Heidrich. *Viewpoint Selection using Viewpoint Entropy*. In Workshop on vision, modelling and visualisation, volume 1, pages 273–280, 2001. (Cité en pages iii, 26, 27 et 30.)
- [Weiss 1987] L. Weiss, A. Sanderson et C. Neuman. *Dynamic sensor-based control of robots with visual feedback*. IEEE Journal of Robotics and Automation, vol. 3, no. 5, pages 404–417, 1987. (Cité en pages 12 et 14.)
- [Wilson 1996] W. Wilson, C. Hulls et G. Bell. *Relative end-effector control using cartesian position based visual servoing*. IEEE Transactions on Robotics and Automation, vol. 12, no. 5, pages 684–696, 1996. (Cité en page 18.)
- [Zhang 2000] Z. Zhang. *A flexible new technique for camera calibration*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 11, pages 1330–1334, 2000. (Cité en page 11.)

Résumé : Dans cette thèse, nous traitons le problème de la navigation dans un environnement virtuel 3D. En particulier, les environnements, sous forme d'un nuage de centaines de millions de points, sont difficiles à appréhender par un novice. La complexité et la richesse en détails du nuage de points 3D de la cathédrale d'Amiens peuvent ainsi entraîner une désorientation et une visualisation non pertinente avec les outils (logiciels et interfaces) existants.

Les contributions de la thèse portent sur le contrôle de caméra virtuelle, automatique et assisté, en exploitant des informations visuelles 2D de l'image et d'autres informations 3D de l'environnement. Pour garantir la pertinence visuelle, nous proposons deux méthodes de guidage de caméra, l'une basée entropie photométrique et la seconde, représentant la contribution majeure de cette thèse, définit et exploite le mélange de gaussiennes basées saillance visuelle. Le formalisme d'asservissement visuel permet de relier la modélisation de l'image aux degrés de liberté de la caméra. L'évitement d'obstacles, la fluidité du mouvement et l'orientation appropriée de la caméra sont des contraintes supplémentaires, prises en compte pour deux types de navigations : le cadrage local et l'exploration globale.

Le but du cadrage visuel est de déplacer la caméra en utilisant la nouvelle primitive photométrique du mélange de gaussiennes basées saillance, afin d'atteindre un point de vue pertinent pour la visualisation d'un objet. L'approche a été testée en simulation, sur un nuage de points dense et aussi dans un environnement réel avec un robot. En ce qui concerne l'exploration, deux méthodes ont été proposées. La première automatique en utilisant l'entropie photométrique et les contraintes de réalisme du mouvement. Le problème est résolu à l'aide d'un algorithme d'optimisation hybride et hiérarchique. Ensuite, une approche d'exploration assistée a été proposée, où, l'utilisateur définit un ensemble de zones à visiter. A ce dernier, s'ajoutent différentes contraintes pour assurer une navigation réaliste et pertinente. La résolution s'appuie sur une loi de contrôle hiérarchique qui utilise le formalisme de redondance. Ces deux méthodes d'exploration ont été testées sur des nuages de points très vastes et complexes, notamment celui de la cathédrale Notre Dame d'Amiens.

Mots clés : Navigation, optimisation, entropie, saillance, mélange de gaussiennes

Towards assistive, relevant and realistic exploration of 3D dense environments

Abstract : In this thesis, we address the issue of navigation in virtual 3D environment. In particular, environments made of hundreds of millions of points, which are difficult to bring under control by a novice. The complexity and the wealth of details of the 3D point cloud of the cathedral of Amiens can result in a disorientation and in an irrelevant visualization with existing tools (interfaces).

The contributions of the thesis deal with automatic or assisted camera control exploiting 2D visual information from the image and other 3D information from the environment. To ensure the visual relevance, we propose two methods to pilot the camera, one based on the photometric entropy and the second representing the major contribution of this thesis, defines and exploits the saliency-based Gaussian mixture. The visual servoing formalism is used to link the image modelling to the camera degrees of freedom. The obstacle avoidance, the fluidity of motion and appropriate camera orientation are considered as additional constraints taken into account in two navigation modes : the local framing and the global exploration.

The goal of visual framing is to move the camera by maximizing the saliency-based Gaussian mixture feature, in order to reach a relevant viewpoint to visualize an object. We test this approach in synthetic model, 3D points cloud model and in a real environment with a robot. Regarding exploration, we present first an automatic camera control exploiting the photometric entropy and some constraints to ensure realistic motion. The problem is solved using an hybrid and hierarchical optimization algorithm. Then, we present a navigation aid system helping the user to explore a part or the whole 3D environment. The system is built using the redundancy formalism taking into account several constraints. These approaches were tested on simple and complex dense 3D points cloud.

Keywords : Navigation, optimization, entropy, saliency, Gaussian mixture
