

Université de Montréal

**Calibration de systèmes de caméras et projecteurs
dans des applications de création multimédia**

par

Lucie Bélanger

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des arts et des sciences
en vue de l'obtention du grade de Maître ès sciences (M.Sc.)
en informatique

décembre, 2009

© Lucie Bélanger, 2009

Université de Montréal
Faculté des arts et des sciences

Ce mémoire intitulé:
**Calibration de systèmes de caméras et projecteurs dans des
applications de création multimédia**

présenté par:
Lucie Bélanger

a été évalué par un jury composé des personnes suivantes:

Max Mignotte
président-rapporteur

Sébastien Roy
directeur de recherche

Jian-Yun Nie
membre du jury

RÉSUMÉ

Ce mémoire s'intéresse à la vision par ordinateur appliquée à des projets d'art technologique. Le sujet traité est la calibration de systèmes de caméras et de projecteurs dans des applications de suivi et de reconstruction 3D en arts visuels et en art performatif.

Le mémoire s'articule autour de deux collaborations avec les artistes québécois Daniel Danis et Nicolas Reeves. La géométrie projective et les méthodes de calibration classiques telles que la calibration planaire et la calibration par géométrie épipolaire sont présentées pour introduire les techniques utilisées dans ces deux projets. La collaboration avec Nicolas Reeves consiste à calibrer un système caméra-projecteur sur tête robotisée pour projeter des vidéos en temps réel sur des écrans cubiques mobiles. En plus d'appliquer des méthodes de calibration classiques, nous proposons une nouvelle technique de calibration de la pose d'une caméra sur tête robotisée. Cette technique utilise des plans elliptiques générés par l'observation d'un seul point dans le monde pour déterminer la pose de la caméra par rapport au centre de rotation de la tête robotisée. Le projet avec le metteur en scène Daniel Danis aborde les techniques de calibration de systèmes multi-caméras. Pour son projet de théâtre, nous avons développé un algorithme de calibration d'un réseau de caméras *wiimotes*. Cette technique basée sur la géométrie épipolaire permet de faire de la reconstruction 3D d'une trajectoire dans un grand volume à un coût minime.

Les résultats des techniques de calibration développées sont présentés, de même que leur utilisation dans des contextes réels de performance devant public.

Mots-clés : calibration de caméra sur tête robotisée, système multi-caméras, système caméra-projecteur, wiimotes, écrans mobiles, projection adaptative, reconstruction 3D, art technologique.

ABSTRACT

This thesis focuses on computer vision applications for technological art projects. Camera and projector calibration is discussed in the context of tracking applications and 3D reconstruction in visual arts and performance art.

The thesis is based on two collaborations with québécois artists Daniel Danis and Nicolas Reeves. Projective geometry and classical camera calibration techniques, such as planar calibration and calibration from epipolar geometry, are detailed to introduce the techniques implemented in both artistic projects. The project realized in collaboration with Nicolas Reeves consists of calibrating a pan-tilt camera-projector system in order to adapt videos to be projected in real time on mobile cubic screens. To fulfil the project, we used classical camera calibration techniques combined with our proposed camera pose calibration technique for pan-tilt systems. This technique uses elliptic planes, generated by the observation of a point in the scene while the camera is panning, to compute the camera pose in relation to the rotation centre of the pan-tilt system. The project developed in collaboration with Daniel Danis is based on multi-camera calibration. For this studio theatre project, we developed a multi-camera calibration algorithm to be used with a wiimote network. The technique based on epipolar geometry allows 3D reconstruction of a trajectory in a large environment at a low cost.

The results obtained from the camera calibration techniques implemented are presented alongside their application in real public performance contexts.

Keywords: pan-tilt camera calibration, multi-camera system, camera-projector system, wiimotes, mobile screens, adaptative projection, 3D reconstruction, technological art.

TABLE DES MATIÈRES

Liste des figures	iii
1 Introduction	1
2 Modélisation de caméras	5
2.1 Géométrie projective	5
2.2 Modèle de caméra	8
2.2.1 Paramètres internes	9
2.2.2 Paramètres externes	10
2.2.3 Distorsion radiale	11
3 Méthodes de calibration de caméra	14
3.1 Calibration planaire	14
3.2 Calibration par géométrie épipolaire	17
3.2.1 Matrice fondamentale \mathcal{F}	18
3.2.2 Matrice essentielle \mathcal{E}	20
3.3 Calibration de la distorsion radiale	21
3.4 Calibration du zoom	23
3.4.1 Calibration de la distorsion radiale	24
3.4.2 Calibration de la matrice de paramètres internes	24
4 Calibration de systèmes caméra-projecteur sur tête robotisée	26
4.1 Utilisation pratique des DLx	29
4.1.1 DMX512	29
4.1.2 Illuminateur infrarouge	30
4.1.3 Filtre de lumière visible	30
4.2 Modélisation des DLx	31
4.3 Calibration de la caméra	31
4.3.1 Calibration des paramètres internes	32
4.3.2 Calibration de la pose relative au monde du DLx	38
4.4 Calibration du projecteur	44
4.4.1 Estimation des paramètres internes	44
4.4.2 Calibration de la pose dans le système du DLx	46
4.5 Correction de la calibration caméra-projecteur par la méthode de minimisation non-linéaire de Levenberg-Marquardt	49

4.6	Résultats	52
4.6.1	Calibration caméra-projecteur	53
4.6.2	Calibration du modèle complet de DLx	54
4.7	Discussion	59
4.8	Application de la calibration	61
5	(ARTICLE) Virtual 3D drawing tool for stage performances	66
5.1	Introduction	70
5.1.1	Technology in theater performances	71
5.1.2	Working with wiimotes	74
5.2	Multiple wiimote calibration	76
5.2.1	Internal calibration	77
5.2.2	Pose from epipolar geometry	78
5.2.3	Pose minimization	79
5.2.4	Unit system transformation	81
5.3	Results	81
5.4	System implementation	86
5.4.1	Data acquisition	86
5.4.2	Calibration program	86
5.4.3	Real-time triangulation	88
5.5	Conclusion	88
6	Conclusion	91
6.1	Applications futures	92
	Références	93
	A Projections adaptatives sur les Tryphons	99
	B Dessin 3D immersif	101

LISTE DES FIGURES

2.1	Représentation des coniques.	5
2.2	Exemples de transformations projectives.	7
2.3	Modélisation d'une caméra à sténopé, d'après Hartley & Zisserman [15].	8
2.4	Relation entre le système de coordonnées de la caméra et la pose de la caméra, d'après Hartley & Zisserman [15].	9
2.5	Effets de la distorsion radiale.	12
2.6	Exemple d'image avec de la distorsion radiale de type <i>barillet</i> . La photo a été prise avec une caméra Prosilica GE650C et une lentille Kowa 4,5mm f/1,4.	13
3.1	Illustration de la géométrie épipolaire, d'après Hartley & Zisserman [15].	18
3.2	Exemple de fonction du facteur de distorsion radiale en fonction du rayon $d(r)$, où $k_1 = -0.2892$ et $k_2 = 0.2568$	23
4.1	Système caméra-projecteur utilisé pour le projet.	26
4.2	Filtre de lumière visible.	31
4.3	Modélisation d'un DLx.	32
4.4	Échantillonnage de la distorsion radiale en observant une ligne. Les images permettent de remarquer le réglage automatique de la caméra et son vignetage.	33
4.5	Valeurs discrètes de k_1 et k_2 trouvées lors de la calibration et leur interpolation respective en fonction de la valeur du zoom DMX. . . .	34
4.6	Fonctions du coefficient de distorsion radiale pour les 37 incréments de k_1 et k_2 calibrés.	34
4.7	Fonction du coefficient de distorsion radiale $d(r, z)$ en fonction du zoom z et du rayon de distorsion r	35
4.8	Patron de la grille de calibration.	36
4.9	Calibration des focales en fonction des incréments de zoom d'une caméra DLx.	37

4.10	Calibration du point principal en fonction des incréments de zoom d'une caméra DLx.	37
4.11	Modélisation du DLx.	39
4.12	Génération de plans elliptiques par la rotation de la caméra autour de l'axe vertical en regardant un point dans le monde (en rouge).	41
4.13	Retrouver le centre du DLx à partir des poses ayant subies une rotation horizontale.	42
4.14	Résultat de la minimisation de la pose de caméra dans le système du DLx.	45
4.15	Exemple de lumière structurée pour les bits 0 à 3 en x	47
4.16	Résultat de la lumière structurée sauvegardé dans une image.	47
4.17	Correspondances extraites de la lumière structurée.	49
4.18	Résultat de la minimisation Levenberg-Marquardt dans la reconstruction des points 3D. Les points originaux sont en bleu et les points résultants de la minimisation sont en rouge.	50
4.19	Erreurs de reprojection des points 3D dans la caméra avant (a) et après (c) la minimisation. Erreurs de reprojection des points 3D dans le projecteur avant (b) et après (d) 20 itérations de minimisation Levenberg-Marquadt.	53
4.20	Visualisation des poses caméra et projecteur et reconstruction 3D du plan de calibration.	54
4.21	Projection de la grille détectée par la caméra et adaptée au point de vue du projecteur.	55
4.22	Reconstruction d'un point 3D à partir d'une correspondance caméra-projecteur et sa reprojection dans le projecteur après avoir subi un déplacement de la tête robotisée.	56
4.23	Reprojection du point 3D de référence dans le projecteur, après avoir appliqué des rotations horizontale et verticale à la tête robotisée.	57
4.24	Détection de la cible dans l'image (noir) et distribution de l'erreur de reprojection du point 3D (gradient mauve).	57

4.25	Visualisation des positions de la caméra (rouge) et du projecteur (orange) avec la reconstruction 3D du point de référence (vert) et les reconstructions 3D des correspondances compensées par le mouvement du DLx (bleu).	58
4.26	Visualisation de la précision de l'intersection des rayons du point de référence 3D.	60
4.27	Visualisation de l'erreur de reconstruction 3D en fonction de la rotation de la tête robotisée (gradient mauve) et l'échantillon de reconstruction (noir).	60
4.28	Positionnement de trois DL1s sur le bord de la piste de danse.	62
4.29	Détection du cube et ajustement de sa pose en utilisant les contours.	63
4.30	Capture d'écran de l'interface. On peut voir que le suivi a été mal initialisé sur les deux premières images, l'interface nous permettait de corriger ce genre de problèmes de suivi pendant la performance.	64
4.31	Projections des vidéos d'Elektra et de Jason E. Lewis sur les faces des <i>Tryphons</i> au Grand Palais de Paris.	65
5.1	Manettes de contrôle de la console Wii de Nintendo (ou <i>wiimotes</i>).	68
5.2	Stage configuration of <i>La Trilogie des Flous</i> as presented at <i>Usine C</i> in Montreal.	72
5.3	Daniel Danis drawing on a circular screen with the tool during a performance of <i>La Trilogie des Flous</i> at <i>Usine C</i> in Montreal.	73
5.4	Custom-made wide-angle infrared pointer using 16 infrared LEDs and two AA batteries.	75
5.5	Infrared visibility of the pen position at angles -90° , -60° , -25° , 0° , 25° , 60° and 90° from the camera optical axis.	75
5.6	Graphs representing the brightness of the IR pen for angles between -130° et 130° (90° angles are represented with orange lines).	76
5.7	Each wiimote's viewing angle was calibrated independently by marking the edges of the field of view on a whiteboard at a known distance.	78
5.8	Even though the camera positions look good, the 3D reconstruction shows major misalignments (shown with arrows).	79

5.9	Illustration of the minimization process. Dark grey = fixed camera, white = minimized camera, light gray = not considered at this step.	82
5.10	Poses obtained after complete calibration seen from two viewpoints. Compared to figure 5.8 the 3D reconstruction is continuous.	83
5.11	Reprojection (in red) of reconstructed 3D points visible from all cameras. Original wiimote coordinates are shown in blue.	84
5.12	Histogram of the reprojection errors (in pixels) illustrating the effect of Ransac.	84
5.13	Example of 3D triangulation showing intersecting rays for a point visible from four wiimotes.	85
5.14	Relative 3D reconstruction error (in cm) for calibration results with 3D points of different visibility.	86
5.15	3D reconstruction on all points with different subsets of cameras.	87
5.16	Examples of different wiimote configurations.	89
A.1	Morphologie d'un <i>Tryphon</i>	99
A.2	Test de projection sur un <i>Tryphon</i> au Coeur des Sciences de l'UQÀM. La vidéo projetée est le projet <i>Cityspeak</i> , de Jason E. Lewis, qui reçoit et affiche en temps réel les messages textes envoyés par des utilisateurs anonymes.	100
B.1	Positionnement des <i>wiimotes</i> autour de l'écran immersif.	101
B.2	Dessin stéréoscopique et visualisation à l'aide de lunettes polarisées.	101
B.3	Tests de dessins 3D immersifs en temps réel.	102

REMARQUES ET ABRÉVIATIONS

- La forme masculine est employée pour désigner à la fois le masculin et le féminin.

actant en théâtre contemporain, entité qui influe sur le cours narratif de la performance (ex : un objet, un personnage, un groupe de personnes, etc.)

DEL (abbrv.) Diode ÉlectroLuminescente

DLx (abbrv.) modèles de projecteur DL1 et/ou DL2 de la compagnie *High End Systems*

DMX (abbrv.) *Digital MultipleX*, protocole de communication des appareils scéniques

idéateur personne à la source d'une idée mais qui ne l'a pas mis en pratique elle-même

SAILS (abbrv.) *Self-Assembling Intelligent Lighter-than-Air Structures*, projet artistique de Nicolas Reeves

Tryphon dernière version des robots du projet SAILS

USITT (abbrv.) *United States Institute for Theatre Technology*

VJ (abbrv.) vidéo-jockey, une personne qui mixe des flux vidéos pour créer des effets vidéos en temps réel qui accompagnent la musique

wimotes (angl.) manettes de la console Wii de la compagnie Nintendo

NOTATION

\mathbf{P}	Point projectif 3D
\mathbf{p}	Point projectif 2D
\mathbf{l}	Ligne
π	Plan
\mathcal{C}	Conique
Ω_∞	Conique absolue
ω	Image de la conique absolue
\mathcal{H}	Transformation projective, homographie
\mathcal{M}	Matrice de caméra
\mathcal{K}	Matrice de paramètres internes d'une caméra
f	Longueur focale (ou focale)
(u, v)	Point principal d'une caméra
\mathcal{P}	Matrice de pose d'une caméra
\mathcal{R}	Matrice de rotation de la pose d'une caméra
\mathbf{C}	Position du centre de la caméra
\mathbf{t}	Vecteur de translation de la pose d'une caméra
\mathbf{e}	Épipole
\mathcal{F}	Matrice fondamentale
\mathcal{E}	Matrice essentielle
\mathcal{T}	Matrice de normalisation

REMERCIEMENTS

Je remercie mon directeur de maîtrise, Sébastien Roy, pour ses conseils, sa disponibilité, son soutien et pour toutes les possibilités de collaboration dans lesquelles il m'a impliquée au cours de mes études. Je veux remercier les membres du laboratoire Vision3D pour leur aide et les discussions enrichissantes.

J'aimerais remercier Daniel Danis pour m'avoir donné l'opportunité de travailler sur le projet de *La Trilogie des Flous* et de m'avoir accordé sa confiance. Je remercie également tous les partenaires qui ont investi temps et argent pour la réalisation du projet : le regroupement Numédiart de la Faculté Polytechnique de Mons (BE), ainsi que l'Usine C et Le Fresnoy, Studio national des arts contemporains (FR). Un merci tout particulier à Thierry Dutoit, professeur à la Faculté Polytechnique de Mons et coordonnateur scientifique du regroupement Numédiart, pour son accueil chaleureux et sa générosité pendant nos séjours.

Je voudrais remercier Nicolas Reeves et les membres de son laboratoire, NXI GESTATIO, de m'avoir accueillie dans leur équipe et de m'avoir permis de travailler sur le projet SAILS.

Finalement, je tiens à remercier le département d'informatique et de recherche opérationnelle (DIRO) de l'Université de Montréal et le Fond québécois de la recherche sur la nature et les technologies (FQRNT) pour le soutien financier qu'ils m'ont accordé pendant mes études.

Chapitre 1

INTRODUCTION

Les domaines des arts visuels et des arts performatifs bénéficient depuis plusieurs années de la démocratisation de la technologie. Il est de plus en plus facile pour les artistes de s'équiper d'ordinateurs, de caméras, de senseurs électroniques et de projecteurs. Cette accessibilité a permis l'émergence de créations artistiques où l'image interagit avec les multiartistes et l'environnement. L'intégration de caméras et d'algorithmes d'analyse d'image dans les systèmes de projection multiplie les possibilités artistiques mais les artistes sont souvent limités par leurs connaissances ou la disponibilité des outils logiciels et ne peuvent tirer pleinement profit de la technologie.

Dans des applications artistiques, l'analyse d'image permet des interactions plus riches entre les actants et le contenu multimédia. Les algorithmes d'analyse d'image extraient des informations d'une image telles que les contours, les formes et la composition fréquentielle. Les techniques de vision par ordinateur analysent cette information à un plus haut niveau et tirent des conclusions sur la nature des objets dans l'image ou sur le mouvement perçu dans une suite d'images. Le système visuel humain est capable d'analyser les contours, de segmenter les différents objets, de les reconnaître, de les suivre et d'extraire l'information 3D de la scène de façon innée. Pour qu'un ordinateur effectue ce genre de tâches par l'entremise de caméras, plusieurs problèmes de modélisation doivent être résolus.

La calibration de caméra consiste à modéliser le système de capture optique d'une caméra. La capture est le processus qui convertit l'information tridimensionnelle du monde en image bidimensionnelle grâce à un senseur numérique et un système de lentilles. La calibration de caméra permet de représenter mathématiquement le système et de générer des images à partir d'informations provenant du monde 3D, ou de déduire la composition du monde 3D à partir d'images ou de séquences d'images. L'analyse d'une scène à l'aide de ces modèles de caméras permet de nombreuses applications telles que la reconstruction 3D, le suivi 3D, l'interpolation de vues et la pro-

jection adaptative. Appliquées dans un contexte artistique, ces techniques permettent de suivre les mouvements d'un acteur, de réaliser des projections architecturales ou de projeter sur des écrans mobiles.

Dans ce mémoire, je m'intéresse aux applications artistiques de la calibration de caméra. La calibration des systèmes optiques utilisés sur scène et dans les installations artistiques améliore la précision du suivi et des projections adaptatives. Un système de caméras calibrées permet d'extraire plus d'information à propos de la structure 3D des objets présents sur la scène (position, forme, grandeur, etc.), ce qui offre plus de possibilités d'interaction avec les artistes. Je présente les travaux résultant de deux importantes collaborations avec les artistes québécois Nicolas Reeves et Daniel Danis. La décision d'incorporer la technologie dans ces deux projets est un choix artistique et esthétique qui ne serait pas réalisable sans l'aide des techniques tridimensionnelles de vision par ordinateur. Ces deux contextes distincts ont grandement influencé le choix de l'équipement utilisé et la démarche scientifique entreprise.

Les Tryphons : collaboration avec Nicolas Reeves

Une première collaboration s'est développée avec Nicolas Reeves, professeur à l'école de design de l'UQÀM, dans le cadre de son projet SAILS (*Self-Aligning-Lighter-than-Air-Structure*). Les *Tryphons* sont des structures cubiques de deux mètres d'arrêtes gonflées à l'hélium (annexe A). Ces robots flottants contrôlent huit micro-turbines qui leur permettent de se mouvoir dans l'espace. Notre contribution au projet consistait à développer un système de suivi et de projections adaptatives en temps réel pour permettre de projeter des vidéos sur les faces visibles des cubes. Pour réaliser ce projet à grande échelle, nous avons utilisé plusieurs systèmes caméra-projecteur sur tête robotisée de la compagnie *High End Systems*¹.

La calibration du système caméra-projecteur nous permet de connaître la position 3D d'un *Tryphon* une fois segmenté dans l'image. La procédure de calibration décrite dans ce mémoire utilise plusieurs méthodes classiques de calibration telles que la calibration par plans et la calibration par géométrie épipolaire. Nous avons développé une méthode de calibration de la pose d'une caméra sur tête robotisée basée sur l'utilisation de plans elliptiques. Cette méthode nous a permis de réaliser la modélisation complète des systèmes caméra-projecteur sur tête robotisée.

¹ High End Systems. Digital Lighting - Products - High End Systems. Dernière visite : 4 octobre 2009. http://www.highend.com/products/digital_lighting/.

La Trilogie des flous : collaboration avec Daniel Danis

Une autre collaboration est née du désir du dramaturge Daniel Danis de combiner théâtre et technologies dans un nouveau genre de performance. En plus d'utiliser des projections adaptatives dans ses performances, Daniel Danis voulait créer un outil qui permettrait à un acteur de dessiner dans l'espace et de visualiser le résultat en temps réel. Pour réaliser un tel outil de dessin 3D, nous avons opté pour un suivi 3D multi-caméras qui utiliserait les caméras infrarouges des manettes de la console Wii de Nintendo (figure 5.1 page 68).

Il existe de nombreuses techniques de calibration développées pour des réseaux de caméras. L'utilisation des *wimotes* comme caméras restreint les possibilités de méthodes de calibration. La calibration par géométrie épipolaire permet de mettre en correspondance les positions des manettes dans l'espace, ce qui permet de réaliser la reconstruction et le suivi 3D. L'algorithme de calibration présenté dans ce mémoire est basé sur des techniques connues de systèmes multi-caméras mais est adapté au contexte théâtral dans lequel il est appliqué.

Organisation du mémoire

Dans un premier temps, j'aborde les notions de géométrie projective nécessaires à la compréhension de ce mémoire. Étant donné qu'il existe plusieurs conventions pour modéliser les éléments d'un système optique, il est essentiel de définir les modèles de caméras utilisés.

Une fois ces bases établies, je présente les méthodes de calibration utilisées dans mes travaux. Les bases théoriques de la calibration planaire sont présentées puisqu'il s'agit de la technique de calibration la plus souvent utilisée dans les applications en vision par ordinateur. J'aborde également la calibration par géométrie épipolaire, très souvent utilisée dans les algorithmes de calibration automatique. Je présente une technique de calibration de distorsion radiale, une déformation de l'image présente dans un grand nombre de caméras. Finalement, je montre comment calibrer et modéliser les paramètres internes d'une caméra à focale variable.

Je détaille l'application de ces techniques de calibration dans le contexte du projet de projection sur les *Tryphons*. J'introduis les particularités techniques des DLx utilisés pour ce projet pour ensuite présenter notre algorithme de calibration par plans

elliptiques. Je conclus cette section par l'application des travaux de calibration dans un contexte réel de performance.

La collaboration avec Daniel Danis en théâtre expérimental est présentée sous la forme d'un article publié dans les actes de conférence de la *Virtual Reality International Conference 2009* (VRIC'09) [4]. L'article présente le contexte artistique dans lequel la technique de calibration a été développée et les étapes nécessaires à la calibration d'un réseau de manettes de Wii pour faire de la reconstruction 3D.

Chapitre 2

MODÉLISATION DE CAMÉRAS

Ce chapitre aborde la notation et les notions de base en géométrie projective nécessaires à la compréhension des techniques de calibration de caméra présentées au chapitre suivant. Les entités géométriques et les particularités des transformations projectives seront présentées. Nous présenterons en détail le modèle de caméra utilisé dans ce mémoire et nous détaillerons les différents paramètres qui le compose.

2.1 Géométrie projective

La géométrie projective étudie les propriétés des entités géométriques et leurs invariances après avoir subi une transformation projective. La géométrie projective se compose de points \mathbf{p} (2D) ou \mathbf{P} (3D), de lignes \mathbf{l} (2D) ou \mathbf{L} (3D), de plans π (3D), de coniques C et de quadriques Q .

Les chapitres suivants supposent que les concepts de point, de ligne et de plan sont connus du lecteur. Quant à la conique C , elle se définit comme une courbe résultant de l'intersection d'un plan avec un cône droit (voir figure 2.1). Une conique se représente

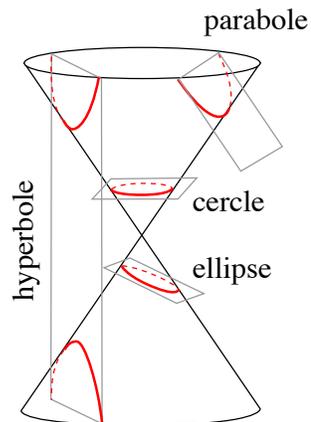


FIG. 2.1 – Représentation des coniques.

en forme matricielle par

$$\mathcal{C} \propto \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}. \quad (2.1)$$

Les points sont exprimés en représentation homogène de forme

$$\mathbf{P} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \equiv \begin{pmatrix} hx \\ hy \\ hz \\ h \end{pmatrix}, \mathbf{P} \in \mathbb{P}^3 \text{ et } (h \neq 0)$$

où h est appelé *coordonnée homogène* ou *coordonnée projective*. Il est possible de passer d'un point euclidien \mathbf{P} à un point projectif \mathbf{P}' par

$$\mathbf{P} \rightarrow \mathbf{P}', \quad \begin{pmatrix} x \\ y \\ z \end{pmatrix} \rightarrow \begin{pmatrix} hx \\ hy \\ hz \\ h \end{pmatrix}, \quad (h \neq 0)$$

et d'un point projectif \mathbf{P}' à un point euclidien \mathbf{P} par

$$\mathbf{P}' \rightarrow \mathbf{P}, \quad \begin{pmatrix} x \\ y \\ z \\ h \end{pmatrix} \rightarrow \begin{pmatrix} x/h \\ y/h \\ z/h \end{pmatrix}.$$

Une transformation projective est une transformation des points et des lignes sur un plan π vers un autre plan π' (figure 2.2). Cette transformation, appelée *homographie*, est la transformation linéaire la plus générale en géométrie projective et englobe toutes les transformations affines. Les transformations affines préservent la collinéarité des points et le ratio des distances, alors que les transformations projectives préservent la collinéarité des points mais pas nécessairement le ratio des distances.

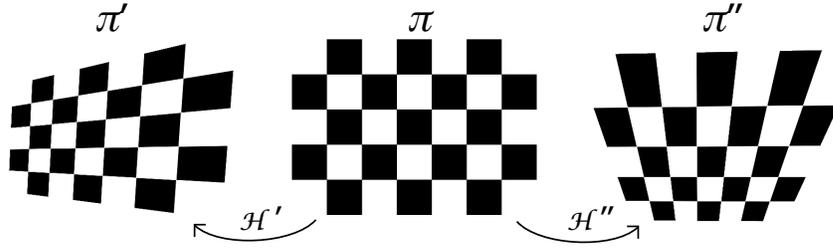


FIG. 2.2 – Exemples de transformations projectives.

L'homographie la plus courante est de forme 2D

$$\mathbf{p} = \mathcal{H}\mathbf{q} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \mathbf{q} \text{ où } \mathbf{q}, \mathbf{p} \in \mathbb{P}^2. \quad (2.2)$$

Les transformations d'un point, d'une ligne et d'une conique par une transformation projective s'écrivent

$$\mathbf{p}' = \mathcal{H}\mathbf{p}, \quad (2.3)$$

$$\mathbf{l}' = \mathcal{H}^{-\top}\mathbf{l}, \quad (2.4)$$

$$\mathcal{C}' = \mathcal{H}^{-\top}\mathcal{C}\mathcal{H}^{-1} \quad (2.5)$$

où \mathbf{p} est un point appartenant à \mathbb{P}^2 , \mathbf{l} définit une ligne telle que

$$\mathbf{l} \begin{pmatrix} x & y & 1 \end{pmatrix} = 0$$

avec

$$\mathbf{l} \propto \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

et \mathcal{C} définit une conique telle que

$$\begin{pmatrix} x & y & 1 \end{pmatrix} \mathcal{C} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}^{\top} = 0.$$

En géométrie projective 2D, les lignes parallèles se rejoignent sur une ligne à l'infini notée \mathbf{l}_∞ et constituée de points à l'infini. De façon similaire en géométrie projective 3D, les lignes et les plans parallèles se rejoignent sur un plan à l'infini notée π_∞ qui ne contient que des points à l'infini. Les points à l'infini 2D et 3D se caractérisent par une coordonnée homogène nulle ($h = 0$) et sont respectivement notés

$$\mathbf{p}_\infty = \begin{pmatrix} x \\ y \\ 0 \end{pmatrix} \quad \text{et} \quad \mathbf{P}_\infty = \begin{pmatrix} x \\ y \\ z \\ 0 \end{pmatrix}.$$

2.2 Modèle de caméra

La calibration de caméra consiste à modéliser le système optique d'une caméra. Le modèle de caméra perspective le plus courant est le modèle à sténopé (*pinhole*), il est à la base de la géométrie projective.

Le modèle de caméra à sténopé se compose d'un centre de caméra \mathbf{C} et d'un plan focal situé à une distance focale f (aussi appelée *focale*) du centre optique. Pour chaque point du monde \mathbf{P} , un point image \mathbf{p} est formé sur le plan focal, à l'intersection du plan focal et du rayon reliant le point \mathbf{P} au centre de caméra \mathbf{C} (figure 2.3). La ligne perpendiculaire au plan focal qui le relie au centre de la caméra est appelée *axe principal*, *rayon principal* ou *axe optique*. Finalement, l'endroit où l'axe optique croise le plan focal s'appelle le *point principal* ou *centre de l'image*.

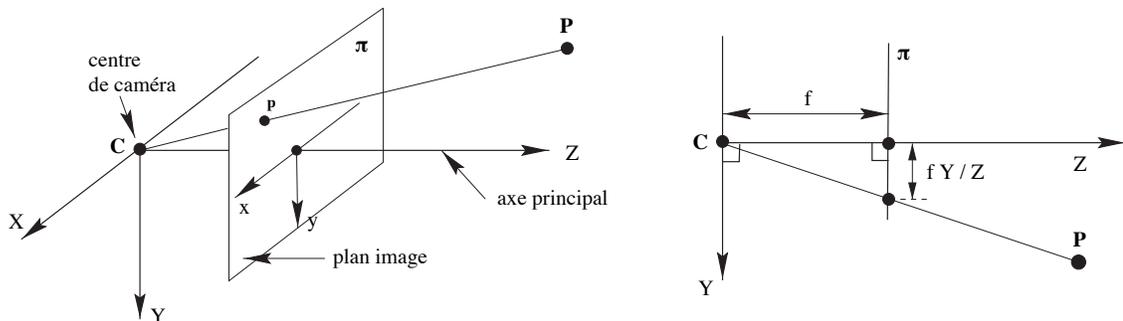


FIG. 2.3 – Modélisation d'une caméra à sténopé, d'après Hartley & Zisserman [15].

La position de la caméra est exprimée selon un système de coordonnées Euclidien dont l'origine correspond au centre de caméra. Par convention, l'orientation du système est canonique, c'est-à-dire que l'axe principal est parallèle à l'axe Z , et que les axes X et Y du système de la caméra sont parallèles aux axes X et Y de l'image. L'origine des points images dans le modèle présenté est situé au coin supérieur gauche (figure 2.4).

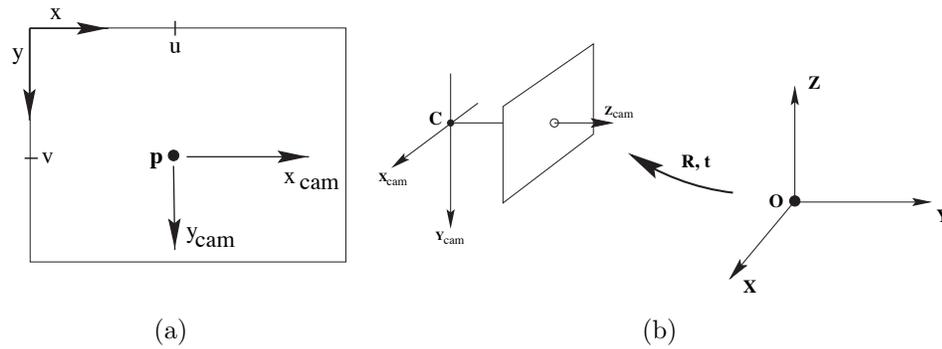


FIG. 2.4 – Relation entre le système de coordonnées de la caméra et la pose de la caméra, d'après Hartley & Zisserman [15].

Il existe plusieurs conventions pour décrire un modèle de caméra. Les termes *point focal*, *centre optique* et *centre de caméra* sont équivalents, ainsi que les termes *plan focal* et *plan image*.

2.2.1 Paramètres internes

Avec le modèle de caméra à sténopé, un point dans le système de coordonnées de la caméra $\mathbf{P}_{cam} = (X, Y, Z)^T$ se projette en \mathbf{p} dans l'image, à l'endroit où le rayon reliant le point \mathbf{P}_{cam} et le centre de caméra \mathbf{C} rencontre le plan image positionné en $Z = f$. Par les propriétés des triangles similaires, on voit que $(X, Y, Z)^T \mapsto (fX/Z, fY/Z)^T$ (figure 2.3). Cette équation suppose que le point principal est situé à l'origine des points images, alors que le point principal se trouve au centre de l'image. Il faut donc ajouter un déplacement $(u, v)^T$ qui représente les coordonnées du point où l'axe optique rencontre le plan image. La projection peut être écrite sous la forme $(X, Y, Z)^T \mapsto (fX/Z + u, fY/Z + v)^T$.

Cette projection peut être exprimée comme une multiplication matricielle en exprimant le point \mathbf{P}_{cam} en coordonnées homogènes.

$$\begin{pmatrix} fX + Zu \\ fY + Zv \\ Z \end{pmatrix} = \begin{bmatrix} f & u & 0 \\ & f & v \\ & & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}. \quad (2.6)$$

Si le point \mathbf{p} est également exprimé sous forme de coordonnées homogènes, $\mathbf{p} = (x, y, 1)^\top$, on peut exprimer l'équation 2.6 sous la forme $\mathbf{p} = \mathcal{K}[\mathbf{I}|0]\mathbf{P}_{cam}$ où

$$\mathcal{K} = \begin{bmatrix} f & u \\ & f & v \\ & & 1 \end{bmatrix} \quad (2.7)$$

est la matrice de paramètres internes. Cette formulation de la matrice de paramètres internes est bonne si les pixels du senseur sont carrés, ce qui est vrai la plupart du temps. Si on suppose que les pixels ne sont pas obligatoirement carrés, les paramètres internes seront exprimés par

$$\mathcal{K} = \begin{bmatrix} f_x & u \\ & f_y & v \\ & & 1 \end{bmatrix} = \begin{bmatrix} f & u \\ & f\alpha & v \\ & & 1 \end{bmatrix} \quad (2.8)$$

où f_x et f_y sont respectivement les focales en x et y et α est un ratio entre f_x et f_y .

La calibration de caméra nous permet de retrouver les longueurs focales f_x , f_y , et le centre de l'image $(u, v)^\top$.

2.2.2 Paramètres externes

La matrice de paramètres internes \mathcal{K} exprime les points 3D selon le système de coordonnées de la caméra, or les points 3D sont habituellement exprimés par rapport à un système de coordonnées qui n'est pas centré sur la caméra. Le système de coordonnées du monde est relié au système de coordonnées de la caméra par une translation \mathbf{t} et une rotation \mathcal{R} . On peut exprimer le point 3D \mathbf{P} dans le système de la caméra par $\mathbf{P}_{cam} = \mathcal{R}(\mathbf{P} - \mathbf{C})$, où \mathbf{C} est la position du centre de la caméra dans le

monde et \mathcal{R} est une matrice de rotation 3×3 . Cette transformation peut être écrite sous la forme homogène

$$\mathbf{P}_{cam} = \begin{bmatrix} \mathcal{R} & -\mathcal{R}\mathbf{C} \\ 0 & 1 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{bmatrix} \mathcal{R} & -\mathcal{R}\mathbf{C} \\ 0 & 1 \end{bmatrix} \mathbf{P}.$$

En incluant les paramètres internes dans la formulation, nous pouvons transformer un point \mathbf{P} du monde en coordonnées image \mathbf{p} par

$$\mathbf{p} = \mathcal{K}\mathcal{R}[\mathbf{I} | -\mathbf{C}]\mathbf{P} \quad (2.9)$$

ou sous la forme plus concise

$$\mathbf{p} = \mathcal{K}[\mathcal{R}|\mathbf{t}]\mathbf{P} \quad (2.10)$$

où $\mathbf{t} = -\mathcal{R}\mathbf{C}$. La calibration des paramètres externes permet de retrouver la position \mathbf{C} et l'orientation \mathcal{R} de la caméra dans le monde.

Les paramètres internes et externes de la caméra sont exprimés dans une même matrice de projection perspective 3×4 ($\mathbb{P}^3 \rightarrow \mathbb{P}^2$) appelée matrice de caméra, \mathcal{M} . La transformation projective entre un point 3D \mathbf{P} et son point image \mathbf{p} est alors simplement exprimée par $\mathbf{p} = \mathcal{M}\mathbf{P}$ avec

$$\mathcal{M} = \mathcal{K}[\mathcal{R}|\mathbf{t}] \quad (2.11)$$

La calibration de la caméra permet de calculer \mathcal{M} en trouvant tout d'abord la matrice de paramètres internes \mathcal{K} puis en estimant directement les transformations affines \mathcal{R} et \mathbf{t} .

2.2.3 Distorsion radiale

Le modèle de caméra à sténopé néglige la réfraction des rayons dans la lentille de la caméra en modélisant la lentille par une simple longueur focale. Ce modèle de lentille simplifié suppose que les propriétés de la géométrie perspective sont respectées. Une de ces propriétés est la préservation de la collinéarité, c'est-à-dire qu'une ligne droite dans le monde est une ligne droite dans l'image. En réalité, les lentilles des caméras ont

souvent des caractéristiques optiques qui ont pour conséquence de déformer l'image. Le modèle à sténopé ne tient pas compte de ces aberrations. La déformation la plus courante est la distorsion radiale. On peut observer deux types de distorsion radiale, la distorsion en coussinet (*pincushion*) ou la distorsion en barillet (*barrel*) (figure 2.5). On retrouve fréquemment cette déformation dans des systèmes optiques à grand angle, ou de moins bonne qualité. Il est possible de calibrer la distorsion radiale pour rectifier les images et ainsi préserver la collinéarité. On peut estimer la distorsion radiale par une fonction polynomiale où le facteur de distorsion à un pixel donné est calculé en fonction de la distance entre le pixel et le centre de distorsion radiale de l'image.

Image distorsionnée en barillet

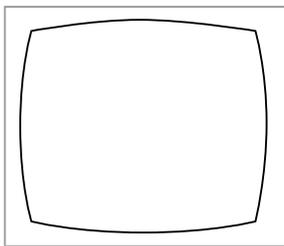


Image linéaire

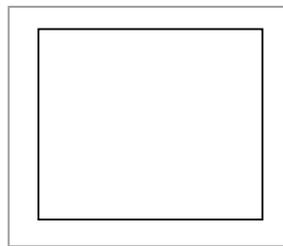


Image distorsionnée en coussinet

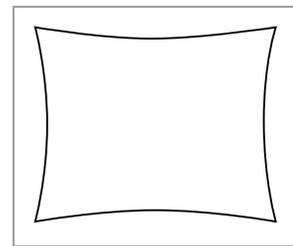


FIG. 2.5 – Effets de la distorsion radiale.

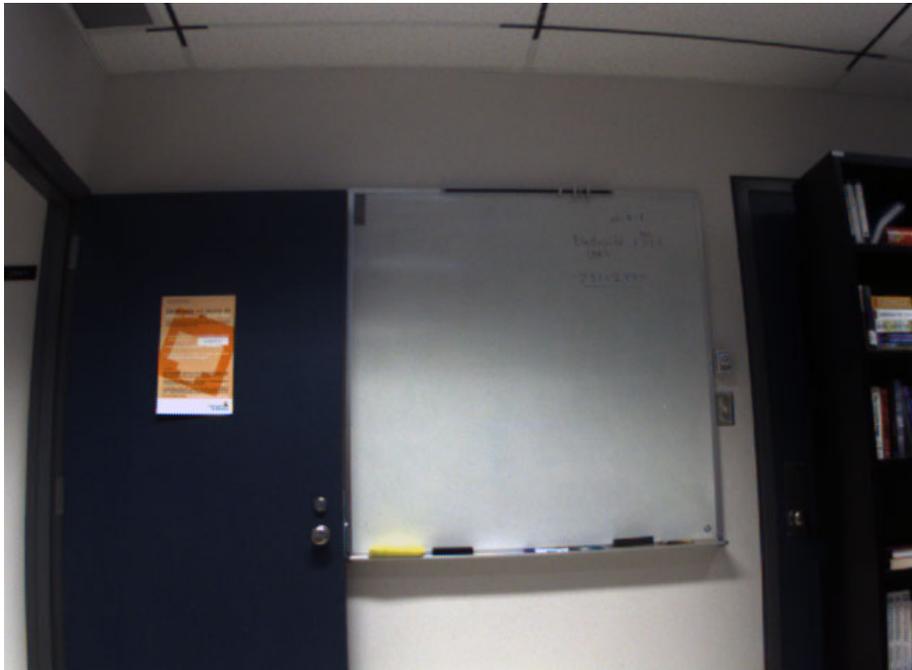


FIG. 2.6 – Exemple d'image avec de la distorsion radiale de type *barillet*. La photo a été prise avec une caméra Prosilica GE650C et une lentille Kowa 4,5mm f/1,4.

Chapitre 3

MÉTHODES DE CALIBRATION DE CAMÉRA

Ce chapitre introduit les différentes approches de calibration de caméra qui nous ont permis d'évaluer les paramètres des caméras à partir d'images. Les techniques de calibration planaire et de calibration par géométrie épipolaire seront présentées. Nous détaillerons la paramétrisation de la distorsion radiale dans un système optique d'une caméra. Finalement, nous présenterons l'application de ces techniques de calibration dans le cas particulier des lentilles à focale variable (zoom).

3.1 Calibration planaire

La calibration planaire est une méthode de calibration de caméra qui permet de déterminer les paramètres internes d'une caméra en utilisant des plans de calibration. Tel que vu à la section 2.1 du chapitre 2, la relation entre deux plans peut s'exprimer par une homographie \mathcal{H} . À partir d'au moins trois plans non-orthogonaux observés par une caméra et les homographies qui relient ces plans au plan image, on peut retrouver les paramètres internes de la caméra en calculant l'image de la conique absolue ω associée aux homographies. Cette méthode a été présentée par Zhang [49] avant de devenir une technique de référence dans le domaine de la calibration de caméra. Les concepts de conique absolue Ω_∞ et d'image de la conique absolue ω seront expliqués plus tard dans ce chapitre.

Tout d'abord, trouvons les transformations projectives $\mathcal{H}_i : \mathbb{P}^2 \rightarrow \mathbb{P}^2$ transformant chaque plan π_i dans le plan image π . Nous savons que deux points correspondants sur des plans sont reliés par

$$\mathbf{p}' = \mathcal{H}\mathbf{p} \tag{3.1}$$

qui peut être exprimé sous la forme

$$\mathbf{p}' \times \mathcal{H}\mathbf{p} = 0. \tag{3.2}$$

À partir de cette contrainte, nous pouvons construire un système d'équations linéaires homogène. Ce système est de forme $\mathbf{Ax} = 0$ où \mathbf{A} contient les coefficients des paramètres de \mathcal{H} dans $\mathbf{p}' \times \mathcal{H}\mathbf{p} = 0$ et \mathbf{x} contient les neuf paramètres de \mathcal{H} . Quatre points de référence suffisent à résoudre le système.

Lors de la calibration par correspondance de points, il est important de normaliser les coordonnées pour augmenter la stabilité numérique du système. La normalisation effectuée sur les données est une mise à l'échelle isotropique telle que décrite dans Hartley & Zisserman [15]. La normalisation consiste à appliquer une transformation \mathcal{T} sur les coordonnées pour que le centroïde des points se retrouve en $(0, 0)$ et que la distance moyenne entre les points et l'origine soit de $\sqrt{2}$. Pour chaque groupe de points \mathbf{p}_i et \mathbf{p}'_i , on trouve respectivement une matrice de normalisation \mathcal{T} et \mathcal{T}' correspondant à

$$\mathcal{T}(\mathbf{p}_i) = \begin{bmatrix} s & 0 & s(-\bar{x}) \\ 0 & s & s(-\bar{y}) \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

où (\bar{x}, \bar{y}) est le centroïde des points originaux et s est le facteur d'échelle qui amène la distance moyenne à $\sqrt{2}$. Une fois que les homographies sur les points normalisés sont calculées, il faut dénormaliser la transformation projective avec

$$\mathcal{H} = \mathcal{T}'^{-1} \tilde{\mathcal{H}} \mathcal{T} \quad (3.4)$$

où $\tilde{\mathcal{H}}$ est l'homographie sur les points normalisés et \mathcal{H} est l'homographie dénormalisée. Les homographies correspondant à chacun des plans de calibration permettent de retrouver l'image de la conique absolue.

Tel qu'introduit au chapitre précédent, une conique \mathcal{C} est une courbe résultant de l'intersection d'un plan avec un cône droit (figure 2.1 page 5) et se représente sous forme matricielle par

$$\mathcal{C} \propto \begin{bmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{bmatrix}. \quad (3.5)$$

La conique absolue, Ω_∞ , est une conique de points complexes dans le plan à l'infini π_∞ . Puisque tous les points de la conique sont situés dans le plan à l'infini, la

projection des points de Ω_∞ est

$$\mathbf{p} = \mathcal{K}\mathcal{R}[\mathbf{I} - \mathbf{C}] \begin{bmatrix} x \\ y \\ z \\ 0 \end{bmatrix} = \mathcal{K}\mathcal{R} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (3.6)$$

Sachant que la transformation projective d'une conique est $\mathcal{C}' = \mathcal{H}^{-\top}\mathcal{C}\mathcal{H}^{-1}$, posant

$$\mathcal{H} = \mathcal{K}\mathcal{R} \quad (3.7)$$

par les équations 3.1 et 3.6 et fixant $\Omega_\infty = \mathbf{I}$, l'image de la conique absolue devient

$$\begin{aligned} \omega &= (\mathcal{H})^{-\top}\mathbf{I}(\mathcal{H})^{-1} \\ &= (\mathcal{K}\mathcal{R})^{-\top}\mathbf{I}(\mathcal{K}\mathcal{R})^{-1} \\ &= \mathcal{K}^{-\top}\mathcal{R}\mathbf{I}\mathcal{R}^{-1}\mathcal{K}^{-1} \\ &= \mathcal{K}^{-\top}\mathcal{K}^{-1} \end{aligned} \quad (3.8)$$

$$\omega \propto \begin{bmatrix} \alpha^2 & 0 & -\alpha^2 u \\ 0 & 1 & -v \\ -\alpha^2 u & -v & f^2\alpha^2 + u^2\alpha^2 + v^2 \end{bmatrix} = \begin{bmatrix} w_1 & 0 & w_4 \\ 0 & w_2 & w_5 \\ w_4 & w_5 & w_3 \end{bmatrix}. \quad (3.9)$$

En géométrie projective, on sait qu'un plan π croise le plan infini π_∞ en une ligne. Cette ligne croise la conique absolue Ω_∞ en deux points qu'on appelle les points circulaires de π . L'image des points circulaires repose sur l'image de la conique absolue ω à l'endroit où la ligne de fuite du plan π croise ω . Nous essayons de trouver ω associée à la projection des points circulaires. L'image des points circulaires est calculée par $\mathcal{H} \cdot [1, \pm i, 0]^\top$, ce qui donne $\mathbf{h}_1 \pm i\mathbf{h}_2$ où \mathbf{h}_1 et \mathbf{h}_2 sont les première et deuxième colonnes de \mathcal{H} . Puisque l'image des points circulaires se trouve sur ω , on obtient

$$(\mathbf{h}_1 \pm i\mathbf{h}_2)^\top \omega (\mathbf{h}_1 \pm i\mathbf{h}_2) = 0$$

ce qui donne deux contraintes linéaires (partie réelle et partie imaginaire)

$$\mathbf{h}_1^\top \omega \mathbf{h}_2 = 0 \text{ et } \mathbf{h}_1^\top \omega \mathbf{h}_1 = \mathbf{h}_2^\top \omega \mathbf{h}_2. \quad (3.10)$$

Les contraintes peuvent être écrites sous la forme

$$\begin{bmatrix} h_1 h_2 & h_7 h_8 & h_2 h_7 + h_1 h_8 & h_5 h_7 + h_4 h_8 \\ h_1^2 - h_2^2 & h_4^2 - h_5^2 & h_7^2 - h_8^2 & 2h_4 h_7 - 2h_5 h_8 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{bmatrix} = 0.$$

Pour chaque plan, on trouve l'homographie entre le plan de calibration et le plan image. En accumulant les contraintes avec un minimum de trois plans, on obtient un système d'équations linéaires homogène de forme $\mathbf{Ax} = 0$ où \mathbf{x} correspond aux paramètres de ω . Chaque plan doit avoir une position et une orientation différentes, et les plans ne doivent pas être orthogonaux. Ce système d'équation peut être résolu par moindres carrés en utilisant une décomposition en valeurs singulières (SVD). On retrouve alors ω à un facteur d'échelle près.

Par l'équation 3.8 nous pouvons retrouver \mathcal{K} avec une décomposition de Cholesky, ou retrouver les paramètres α, f, u et v par calcul direct grâce à l'équation 3.9. Il est possible de retrouver la pose de la caméra par rapport à un des plans observés. En supposant que les points situés sur le plan choisi sont en $z = 0$ dans

$$\mathbf{p} = \mathcal{K}[\mathcal{R}|\mathbf{t}] \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \mathcal{K}[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}] \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \quad (3.11)$$

on obtient

$$\mathcal{K}^{-1}\mathcal{H} = [\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{t}], \quad (3.12)$$

ce qui nous permet de retrouver \mathcal{R} sachant que $[\mathbf{r}_1 \quad \mathbf{r}_2 \quad \mathbf{r}_1 \times \mathbf{r}_2]$ est une matrice de rotation si son déterminant est 1. Puisque nous connaissons maintenant \mathcal{H}, \mathcal{K} et \mathcal{R} , nous pouvons retrouver la translation \mathbf{t} par l'équation 3.12.

3.2 Calibration par géométrie épipolaire

La géométrie épipolaire définit la relation entre deux images prises de point de vue différents, et implique au minimum une translation entre les deux poses de caméras.

Elle est basée sur l'observation qu'un point dans une image se projette à quelque part sur une ligne dans l'autre vue. Plus formellement, un point image \mathbf{p} se déprojette dans l'espace comme un rayon passant par le centre de la caméra \mathbf{C} et le point \mathbf{p} . La projection de ce rayon dans la deuxième caméra donne une ligne l' , appelée *ligne épipolaire*. Pour un point 3D \mathbf{P} situé sur le rayon de projection issu de \mathbf{p} , son image se trouve nécessairement sur l' . Lorsque deux caméras observent le même point 3D \mathbf{P} , les centres \mathbf{C} et \mathbf{C}' des caméras, le point \mathbf{P} et ses images \mathbf{p} et \mathbf{p}' forment un plan appelé *plan épipolaire* (voir figure 3.1(a)). Les deux endroits où la ligne reliant les deux centres de caméra croise les plans images sont appelés *épipoles* et sont notés \mathbf{e} et \mathbf{e}' (voir figure 3.1(b)). Toutes les lignes et les plans épipolaires possibles passent par les épipoles.

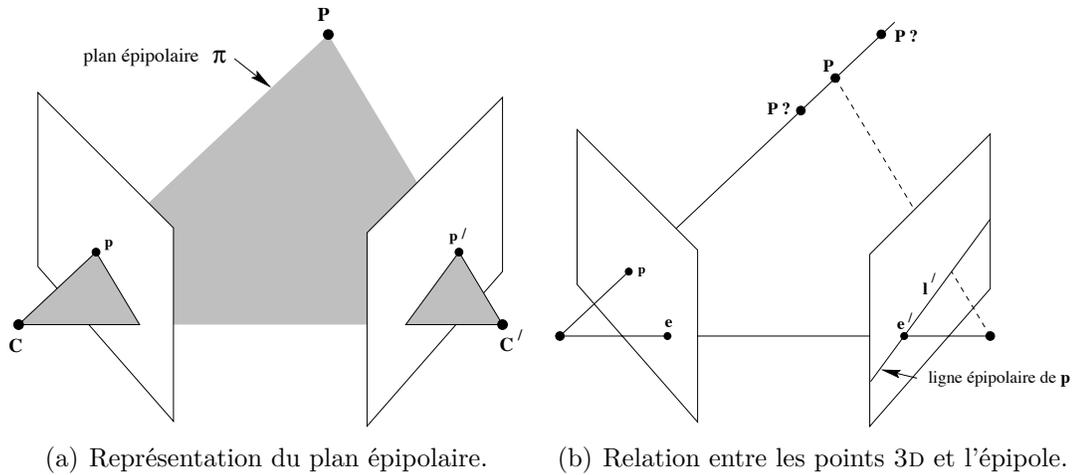


FIG. 3.1 – Illustration de la géométrie épipolaire, d'après Hartley & Zisserman [15].

3.2.1 Matrice fondamentale \mathcal{F}

La matrice fondamentale est une représentation matricielle 3×3 de la relation épipolaire entre deux vues. Il est possible de retrouver \mathcal{F} à partir d'au moins huit correspondances de points \mathbf{p} et \mathbf{p}' . On utilise l'algorithme des huit points normalisés présenté par Hartley [14] suite aux travaux de Longuet-Higgins [23]. La déprojection d'un point image \mathbf{p} à une distance λ sur le rayon de déprojection est donnée par $\mathbf{P}(\lambda) = \mathcal{M}^\dagger \mathbf{p} + \lambda \mathbf{C}$, où \mathcal{M}^\dagger est la pseudo-inverse de la matrice caméra \mathcal{M} . On connaît deux points sur le rayon de déprojection, $\mathcal{M}^\dagger \mathbf{p}$ et \mathbf{C} . En projetant les deux

points dans la deuxième caméra, on obtient une ligne \mathbf{l}'

$$\begin{aligned}\mathbf{l}' &= \overbrace{(\mathcal{M}'\mathbf{C})}^{\mathbf{e}'} \times (\mathcal{M}'\mathcal{M}'^\dagger\mathbf{p}) \\ &= \mathbf{e}' \times (\mathcal{M}'\mathcal{M}'^\dagger\mathbf{p}) \\ &= [\mathbf{e}']_{\times} \mathcal{M}'\mathcal{M}'^\dagger\mathbf{p}.\end{aligned}$$

Puisque la correspondance de \mathbf{p} , \mathbf{p}' se trouve sur la ligne \mathbf{l}'

$$\begin{aligned}\mathbf{p}'^\top \mathbf{l}' &= 0 \\ \mathbf{p}'^\top \underbrace{[\mathbf{e}']_{\times} \mathcal{M}'\mathcal{M}'^\dagger}_{\mathcal{F}} \mathbf{p} &= 0\end{aligned}$$

ce qui donne la contrainte

$$\mathbf{p}'^\top \mathcal{F} \mathbf{p} = \begin{pmatrix} x' & y' & 1 \end{pmatrix} \mathcal{F} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0 \quad (3.13)$$

où la matrice \mathcal{F} exprime la relation entre les deux caméras en reliant les points images \mathbf{p} et \mathbf{p}' . Les points projectifs \mathbf{p} et \mathbf{p}' sont définis à partir de points euclidiens de forme (x, y) et sont exprimés par $\mathbf{p} = (x, y, 1)^\top$ et $\mathbf{p}' = (x', y', 1)^\top$.

Il est possible de retrouver \mathcal{F} avec huit correspondances de points en construisant un système d'équations linéaires homogène de forme $\mathbf{A}\mathbf{x} = 0$ où \mathbf{A} est de forme

$$\mathbf{A} = \begin{bmatrix} xx' & xy' & x & yx' & yy' & y & x' & y' & 1 \\ \vdots & \vdots \end{bmatrix} \quad (3.14)$$

et \mathbf{x} est le vecteur des paramètres de \mathcal{F} .

Comme pour la transformation projective, les groupes de points correspondants doivent tout d'abord être normalisés pour assurer la résolution du système. La même mise à l'échelle est effectuée sur chaque groupe de points (voir équation 3.3) et la dénormalisation de la matrice fondamentale est

$$\mathcal{F} = \mathcal{T}'^{-1} \tilde{\mathcal{F}} \mathcal{T} \quad (3.15)$$

où $\tilde{\mathcal{F}}$ est la matrice fondamentale calculée avec les points normalisés, et \mathcal{F} est la matrice fondamentale finale qui exprime la relation entre les deux caméras. La relation \mathcal{F} inclue les paramètres internes des deux caméras ainsi que les paramètres externes de leur position et orientation relatives.

3.2.2 Matrice essentielle \mathcal{E}

La matrice essentielle représente la différence de pose entre deux caméras, contrairement à la matrice fondamentale qui inclue la pose et les paramètres internes. Lorsque les paramètres internes des caméras sont connus, il est possible de retrouver la matrice essentielle \mathcal{E} et d'en extraire l'information de pose. On peut déterminer la matrice essentielle à partir des points normalisés, ou à partir de la matrice fondamentale. Il est possible de déterminer directement les neuf paramètres de \mathcal{E} par l'algorithme des 8-points de Longuet-Higgins [23] sans passer par la matrice fondamentale. L'idée d'utiliser la matrice fondamentale comme point de départ pour déterminer la pose avec la matrice essentielle a été introduite par Zhang [47] pour rendre l'estimation de pose plus robuste.

La matrice essentielle est reliée à la matrice fondamentale par

$$\mathcal{E} = \mathcal{K}'^T \mathcal{F} \mathcal{K} \quad (3.16)$$

En supposant que la première caméra est placée à l'origine, $\mathcal{M} = [\mathbf{I} \ \mathbf{0}]$, on peut retrouver la pose de la deuxième caméra à un facteur d'échelle près et à une ambiguïté de pose quadruple (deux rotations et deux translations possibles). À partir de la représentation en valeurs singulières de $\mathcal{E} = \mathcal{U} \text{diag}(1, 1, 0)^T \mathcal{V}^T$, on peut dériver deux rotations et deux translations possibles ce qui donne les quatres poses suivantes

$$\begin{aligned} \mathcal{M}' &= [\mathcal{U} \mathcal{W} \mathcal{V}^T | + \mathbf{u}_3] \\ \mathcal{M}' &= [\mathcal{U} \mathcal{W} \mathcal{V}^T | - \mathbf{u}_3] \\ \mathcal{M}' &= [\mathcal{U} \mathcal{W}^T \mathcal{V}^T | + \mathbf{u}_3] \\ \mathcal{M}' &= [\mathcal{U} \mathcal{W}^T \mathcal{V}^T | - \mathbf{u}_3] \end{aligned} \quad (3.17)$$

où \mathbf{u}_3 est la dernière colonne de \mathcal{U} et \mathcal{W} est une matrice orthogonale

$$\mathcal{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Pour déterminer la bonne pose, il suffit de reconstruire les points 3D et de choisir la pose pour laquelle tous les points 3D se situent devant les deux caméras.

Contrairement à la calibration planaire où des plans de calibration doivent être utilisés, la calibration par géométrie épipolaire ne requiert pas de connaissance à priori du monde. Il est possible de déduire les correspondances de points en utilisant des algorithmes de mise en correspondance d'images tels que l'algorithme *SIFT* (*Scale-invariant feature transform*) [24]. La géométrie épipolaire est fréquemment utilisée dans les systèmes de calibration automatique de caméra.

3.3 Calibration de la distorsion radiale

La distorsion radiale est courante dans les systèmes optiques. Le phénomène est souvent visible sur le pourtour des images, où la déviation des rayons est la plus remarquée. Même si la distorsion semble minimale à l'œil nu, elle peut induire une erreur importante dans la calibration. Dans des systèmes multi-caméras par exemple, la distorsion radiale nuit considérablement à la mise en correspondance de points puisque ceux-ci se retrouvent souvent dans les extrémités des images (à cause du faible recouvrement des images). Il est donc important de modéliser la distorsion radiale lors de la calibration de caméra puisque notre modèle de caméra est linéaire et n'en tient pas compte.

La correction de la distorsion radiale consiste à rectifier la position des pixels dans l'image pour compenser la déviation des rayons dans le système optique. La distorsion radiale peut être modélisée par un centre de distorsion et un facteur de distorsion. Le facteur de distorsion radiale est représenté sous forme d'une fonction polynomiale dépendante de la distance radiale entre un pixel et le centre de distorsion.

La modélisation de la distorsion radiale a la forme

$$\begin{pmatrix} x \\ y \end{pmatrix} = d(r) \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} \quad (3.18)$$

où $(x, y)^\top$ est la position d'un pixel dans l'image distorsionnée, $(\hat{x}, \hat{y})^\top$ est la position du pixel rectifié, r est la distance radiale et $d(r)$ est le facteur de distorsion exprimé sous la forme $d(r) = 1 + k_1 r + k_2 r^2 + k_3 r^3 + \dots$. La distance radiale est exprimée par

$$r = \sqrt{(x - u)^2 + (y - v)^2}. \quad (3.19)$$

et $(u, v)^\top$ est le centre de distorsion. Le modèle de distorsion radiale utilisé dans ce mémoire considère seulement les coefficients de puissance paire. La fonction s'écrit donc

$$d(r) = 1 + k_1 r^2 + k_2 r^4 \quad (3.20)$$

La calibration consiste à retrouver le centre de distorsion $(u, v)^\top$ et les coefficients (k_1 et k_2) de la fonction polynomiale de distorsion radiale $d(r)$.

La méthode utilisée pendant les travaux de ce mémoire est celle décrite par Tardif, Sturm & Roy [40], où la distorsion radiale est modélisée comme une fonction de la distance focale par rapport à la distance radiale. La formulation présentée est similaire à la formulation générale, à un facteur d'échelle près correspondant à la distance focale.

Cette méthode utilise l'observation de lignes droites dans l'image et suppose que le centre de distorsion est initialisé au centre de l'image (il peut être ajusté par la suite). Par la définition de la géométrie projective, si un triplet de points est collinéaire dans le monde, sa projection dans l'image, $\{(x_0, y_0)^\top, (x_1, y_1)^\top, (x_2, y_2)^\top\}$, doit être collinéaire. Quand une distorsion radiale est présente dans un système optique, la contrainte de collinéarité n'est pas respectée, sauf si les points sont collinéaires au centre de distorsion. Pour chaque point, nous voulons trouver les facteurs $d(r_0)$, $d(r_1)$ et $d(r_2)$ qui permettent aux points images de respecter la contrainte de collinéarité

$$\begin{vmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ d(r_0) & d(r_1) & d(r_2) \end{vmatrix} = 0 \quad (3.21)$$

ou sous la forme étendue

$$d(r_0) \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} + d(r_1) \begin{vmatrix} x_2 & x_0 \\ y_2 & y_0 \end{vmatrix} + d(r_2) \begin{vmatrix} x_0 & x_1 \\ y_0 & y_1 \end{vmatrix} = 0.$$

Les paramètres k_1 et k_2 sont contraints linéairement par $d(r)$ dans l'équation 3.20 et par l'équation 3.21. En accumulant les contraintes linéaires avec plusieurs triplets de points, on obtient un système d'équations linéaires de forme $\mathbf{Ax} = 0$ où \mathbf{x} correspond au vecteur des paramètres k_1 et k_2 . Ce système d'équation peut être résolu par moindres carrés en utilisant une décomposition en valeurs singulières (SVD). La figure 3.2 montre un exemple de fonction de distorsion radiale. On peut voir que la distorsion radiale est moins importante au centre (pour un petit r).

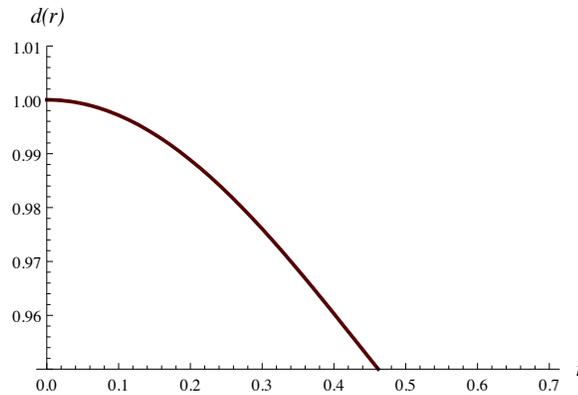


FIG. 3.2 – Exemple de fonction du facteur de distorsion radiale en fonction du rayon $d(r)$, où $k_1 = -0.2892$ et $k_2 = 0.2568$.

Une fois les coefficients de la fonction de distorsion radiale retrouvés, l'image peut être corrigée en calculant les coordonnées rectifiées $(\hat{x}, \hat{y})^T$ (équation 3.18).

3.4 Calibration du zoom

Un objectif à focale variable permet de zoomer à l'intérieur d'une scène pour obtenir plus de détails sur une portion de la scène. La variation de la distance focale est possible grâce au déplacement mécanique d'une lentille à l'intérieur de l'objectif. Le déplacement de la lentille parallèlement à l'axe optique de la caméra modifie l'influence de la distorsion radiale dans l'image, la distance focale ainsi que la posi-

tion du point principal. Il est possible de calibrer l'influence du zoom pour ces trois paramètres.

En calibrant les paramètres internes sur plusieurs incréments de l'intervalle du zoom, on peut construire des fonctions d'interpolation qui nous permettent d'estimer les paramètres de la caméra pour toutes les valeurs du zoom. L'échantillonnage utilisé pour la calibration devrait couvrir tout l'intervalle de zoom pour assurer la qualité de l'interpolation et éviter les erreurs liées à l'extrapolation des valeurs.

3.4.1 Calibration de la distorsion radiale

La distorsion radiale est une aberration optique qui influence principalement les rayons qui sont réfractés sur le pourtour des lentilles. En augmentant la longueur focale, on diminue l'angle de vue de l'objectif. Les rayons se concentrent vers le centre de la lentille et sont beaucoup moins influencés par la distorsion radiale. La distorsion radiale est déjà représentée comme une fonction polynomiale du rayon r (voir l'équation 3.20). En calibrant sur plusieurs incréments de zoom z , nous pouvons exprimer la distorsion radiale comme une fonction de r et z . La fonction polynomiale représentant une courbe d'interpolation permettra d'exprimer les coefficients k_1 et k_2 en fonction de la valeur de l'incrément de zoom. La fonction $d(r)$ devient $d(r, z)$.

3.4.2 Calibration de la matrice de paramètres internes

Pour déterminer la matrice de paramètres internes à chaque intervalle de zoom, il n'est pas nécessaire de calculer l'image de la conique absolue associée à chaque image de plan (voir section 3.1). Une fois les paramètres internes et la pose de la caméra calibrés au zoom 0 à l'aide de la technique de calibration planaire, il est possible de déduire les paramètres internes de la caméra pour tous les autres incréments de zoom. Puisque nous connaissons la transformation projective \mathcal{H} qui relie les points 3D du plan de calibration aux points 2D du plan de caméra, nous pouvons déduire

$\mathcal{K} = \mathcal{H}[\mathcal{R}|\mathbf{t}]^{-1}$ par la relation 2.10 :

$$\begin{aligned}\mathbf{p} &= \mathcal{H}\mathbf{P} \\ \mathbf{p} &= \mathcal{K}[\mathcal{R}|\mathbf{t}]\mathbf{P} \\ \mathcal{H} &= \mathcal{K}[\mathcal{R}|\mathbf{t}] \\ \mathcal{H}[\mathcal{R}|\mathbf{t}]^{-1} &= \mathcal{K}\end{aligned}$$

Puisque la pose $[\mathcal{R}|\mathbf{t}]$ de la caméra ne change pas et que nous connaissons \mathcal{H} pour chaque incrément de zoom à l'aide de la grille de calibration, nous pouvons retrouver les paramètres internes \mathcal{K} pour chaque incrément de zoom, et extraire les valeurs des longueurs focales et du centre optique.

Calibration du déplacement de la focale

La variation de la longueur focale en fonction d'un incrément mécanique n'est pas linéaire. L'échantillonnage des longueurs focales permet de déterminer une fonction polynomiale qui interpole les focales en fonction du zoom. Pour chaque homographie trouvée, on extrait la focale de la matrice \mathcal{K} trouvée avec l'équation 3.22.

Calibration du déplacement du point principal

Pendant l'ajustement du zoom, la lentille se déplace parallèlement à l'axe optique de la caméra. Étant donné que l'alignement des lentilles est rarement parfait, il est fréquent de remarquer que le déplacement de la lentille modifie la position du centre de l'image. L'échantillonnage des paramètres internes permet de modéliser ce déplacement en déterminant une fonction d'interpolation du déplacement en x et en y . Il est important de calibrer les déplacements en x et en y séparément puisqu'ils ne sont pas corrélés. Le centre $(u, v)^T$ est extrait de la matrice \mathcal{K} obtenue avec l'équation 3.22.

Le chapitre suivant présente un exemple de calibration complète d'une lentille à focale variable, à la section 4.3.1.

Chapitre 4

CALIBRATION DE SYSTÈMES CAMÉRA-PROJECTEUR SUR TÊTE ROBOTISÉE

Le projet SAILS a débuté en 2004 dans le laboratoire de design de Nicolas Reeves à l'UQÀM. Les *Tryphons* sont l'implémentation la plus imposante du projet jusqu'à maintenant, avec leur structure cubique de deux mètres d'arrête. Les robots volants sont munis de micro-turbines et de senseurs de distance à ultrasons qui leur permettent de contrôler leurs déplacements en vol. Une fois les *Tryphons* construits, il s'est avéré que les faces des cubes étaient particulièrement appropriées pour recevoir de la projection. Projeter des images et des vidéos sur des écrans cubiques en mouvement nécessite des connaissances approfondies en vision par ordinateur, d'où est née la collaboration entre le laboratoire Vision3D de Sébastien Roy et le laboratoire NXI Gestatio de Nicolas Reeves.



FIG. 4.1 – Système caméra-projecteur utilisé pour le projet.

Un système externe aux cubes utilisant des caméras et des projecteurs a été développé pour suivre les cubes et projeter des vidéos en temps réel sur leurs faces pendant leurs déplacements. Puisque les cubes sont de grande dimension et qu'ils ont une grande amplitude de mouvement, des caméras et des projecteurs fixes ne sont pas appropriés. Nous avons donc utilisé les systèmes caméra-projecteur sur tête robotisée

DL1¹ et DL2² de *High End Systems* (figure 4.1). La robotisation des DLx permet de faire du suivi actif, et leur projecteur puissant permet des projections sur une grande distance. On retrouve fréquemment ces systèmes dans des salles de spectacles. Ils sont très connus sur la scène culturelle mais ne sont malheureusement pas utilisés à leur plein potentiel étant donné les connaissances en vision par ordinateur que cela requiert.

Les systèmes de caméra-projecteur robotisés permettent une plus grande liberté de projection et d'interactions que les projecteurs conventionnels. L'arrivée de ce genre de systèmes sur le marché est justifiée par les nombreuses applications en réalité augmentée. Ces applications facilitent la navigation dans un environnement [7] et créent de nouvelles façons d'interagir avec celui-ci [16, 12].

La calibration de caméras sur tête robotisée nous fournit un point de départ pour la calibration d'un système caméra-projecteur sur tête robotisée. Les systèmes de caméras sur tête robotisée sont fréquemment utilisés dans les applications de vidéo-surveillance [9, 41, 33, 11]. Le mouvement de la caméra permet de couvrir un plus grand champ de vision en utilisant moins de caméras. Dans le contexte de la vidéosurveillance, la calibration est souvent facilitée par un modèle de caméra centré sur son axe de rotation [3, 9, 41, 33, 21]. Pour les applications en vidéosurveillance, l'hypothèse que la caméra est parfaitement centrée est souvent suffisante puisque la translation de la caméra est négligeable par rapport à l'échelle du monde dans lequel le suivi est effectué. Cette hypothèse ne peut être faite avec les systèmes robotisés de *High End Systems* puisque la caméra est clairement décentrée pour laisser place au projecteur. Davis et Chen [11] utilisent un modèle décentré de caméra robotisée similaire au nôtre. La calibration du déplacement de la caméra par rapport aux axes horizontal et vertical est facilitée par l'utilisation d'une cible virtuelle qui permet une plus grande couverture de mouvement lors de la cueillette des données. Un grand angle de rotation permet une calibration plus fiable de la pose de la caméra par rapport au centre de rotation. L'utilisation d'un réseau de caméras fixes pré-calibrées permet d'obtenir des coordonnées 3D pour un grand volume de reconstruction, ce qui donne une plus grande liberté de mouvement à la caméra robotisée et ainsi, une

¹ High End Systems. DL.1 Digital Light - Digital Lighting - Products - High End Systems. Dernière visite : 5 octobre 2009. http://www.highend.com/products/digital_lighting/dl_1.asp.

² High End Systems. DL.2 Digital Light - Digital Lighting - Products - High End Systems. Dernière visite : 5 octobre 2009. http://www.highend.com/products/digital_lighting/dl_2.asp.

meilleure calibration. Étant donné que l'application de nos travaux n'est pas en vidéosurveillance, nous n'avons pas accès à un aussi grand nombre de caméras pour faciliter la calibration.

La calibration de projecteurs sur tête robotisée se rapproche beaucoup de nos travaux. Bien que la plupart des systèmes de projection modulable utilisent des miroirs rotatifs pour déplacer la projection plutôt que de contrôler la pose du projecteur directement [1, 16], les méthodes de calibration pourraient être applicables aux DLx. La modélisation du système optique présentée par Ashdown & Sato [1] ne suppose pas l'alignement des centres de rotation avec le centre optique du projecteur. Leur procédure de calibration est basée sur une calibration planaire classique. La calibration complète par rapport aux centres de rotation est obtenue par la minimisation de l'erreur de reprojection des pixels dans la caméra. Contrairement aux DLx, la caméra utilisée pour la calibration est fixe et n'est pas influencée par le mouvement du miroir.

Peu de littérature concernant la calibration de systèmes caméra-projecteur robotisés a été écrite. Bien que plusieurs projets en réalité augmentée utilisent ce genre de système de vision, la plupart comportent des étapes d'alignement manuel du système de vision [12, 16, 7]. Suite aux travaux du centre de recherche d'IBM sur le *Everywhere Display* [29, 30, 16, 28], Jochen Ehnes, Koichi Hirota et Michitaka Hirose ont tenté le même genre de travail mais avec une modélisation complète du système caméra-projecteur décentré [12]. Leur processus de calibration inclue une mesure manuelle de la distance entre la caméra et le projecteur. Or, il est difficile de déterminer précisément la position du centre optique d'une caméra ou d'un projecteur ; la mesure peut donc être facilement erronée. Notre procédure de calibration ne comporte pas ce genre de manipulation manuelle. Bien que certains paramètres soient estimés, ceux-ci sont corrigés ultérieurement dans des étapes d'ajustement par minimisation non-linéaire.

Les DLx utilisent un système de communication bien détaillé qui permet de contrôler avec précision les mouvements robotiques de la tête et qui inclut quelques contrôles pour la caméra, le projecteur et l'illuminateur infrarouge. Nous avons exploité les avantages de ces contrôles dans le processus de calibration des DLx. La calibration des DLx s'est faite en plusieurs étapes. Tout d'abord, il est important de définir les différents systèmes de coordonnées utilisés pour la calibration des DLx. La calibration complète de la caméra a été effectuée en modélisant la distorsion radiale, les para-

mètres internes, le zoom et la pose de la caméra par rapport au DLx. En utilisant la calibration de la caméra et en estimant les paramètres internes du projecteur, nous avons calibré la pose du projecteur par rapport au DLx.

4.1 Utilisation pratique des DLx

Les systèmes DLx sont conçus pour les applications scéniques. Ils peuvent être complètement contrôlés et programmés par le protocole DMX512³, très souvent utilisé pour les systèmes d'éclairage. La caméra installée sur la tête robotisée permet à l'opérateur de voir l'environnement et de viser la scène avec le projecteur avant d'ouvrir l'iris de projection. Suivant le même raisonnement, l'illuminateur infrarouge permet à l'opérateur de distinguer la scène même si elle est plongée dans le noir complet.

4.1.1 DMX512

Le protocole DMX512 est un standard développé par l'*United States Institute for Theatre Technology* (USITT)⁴ pour contrôler les gradateurs des systèmes d'éclairage. La popularité du protocole auprès des fabricants d'équipements scéniques a fait qu'il est maintenant utilisé pour contrôler des équipements d'effets spéciaux, comme des machines à fumée, et des systèmes d'éclairage plus sophistiqués tels que les projecteurs mobiles de *High End Systems*.

Le protocole permet d'envoyer jusqu'à 512 octets d'information à des dispositifs DMX connectés en guirlande. Chaque dispositif possède un ensemble de commandes pré-définies associées à chaque octet qu'on appelle *canal*. Un canal peut contenir l'information pour plusieurs commandes. Chaque commande est alors assignée à un intervalle de valeurs compris dans l'octet. Pour les commandes nécessitant plus de précision, les valeurs peuvent utiliser plus d'un canal. Le protocole DMX est un mode de transmission arithmique ; chaque trame de données DMX est précédée et terminée par des octets distincts qui représentent le début et la fin du message.

³ USITT DMX512. Dernière visite : 23 mars 2009. <http://www.usitt.org/standards/DMX512.html>.

⁴ USITT. Dernière visite : 23 mars 2009. <http://www.usitt.org/>.

Les DL1 et les DL2 possèdent respectivement 20 et 170 canaux de contrôle. Ils partagent un ensemble commun de commandes dans les dix premiers canaux représentant les contrôles du mouvement de la tête et de l'optique du projecteur. Les canaux suivants sont différents puisqu'ils sont spécifiques aux fonctions particulières des deux modèles. Pour communiquer en DMX avec les DLx, nous utilisons un convertisseur USB à DMX512 qui nous permet de contrôler les DLx à partir d'un ordinateur plutôt qu'à partir d'une console DMX. Ce dispositif rend l'installation plus flexible et plus portable puisqu'il évite d'avoir à programmer une console dédiée aux projecteurs.

4.1.2 *Illuminateur infrarouge*

La lumière infrarouge est fréquemment utilisée dans les applications interactives puisqu'elle est beaucoup plus facile à contrôler dans un endroit public que la lumière ambiante. Elle permet de faire des applications de suivi plus robustes. Dans le cas de projections adaptatives sur des surfaces suivies en temps réel, il est essentiel de faire l'analyse d'image dans le spectre infrarouge pour éviter que les projections n'entraînent une rétroaction dans le suivi. Les DLx sont munis d'un illuminateur à DELs infrarouges et d'une caméra sensible à l'infrarouge. L'intensité de l'illuminateur infrarouge est contrôlable par DMX.

4.1.3 *Filtre de lumière visible*

La caméra est munie d'un filtre infrarouge amovible commandée par DMX. Elle peut percevoir, ou non, la lumière infrarouge. Par contre, elle est toujours sensible au spectre de lumière visible. Cela implique que la caméra des DLx voit toujours les projections, ce qui complique les applications de suivi optique. Pour éviter un problème de rétroaction dans le suivi, nous ajoutons un filtre de lumière visible sur la caméra.

Bien qu'il existe des filtres de lumière visible sur le marché, nous créons nos propres filtres à l'aide de pellicule photographique. L'épreuve négative d'une pellicule couleur exposée à la lumière visible, mais non exposée à l'infrarouge, bloque les longueurs d'ondes correspondant au spectre visible mais laisse passer l'infrarouge.

Chaque caméra de DLx utilisée dans le projet des *Tryphons* est recouverte d'un de ces filtres, ce qui permet de tirer avantage de l'illuminateur infrarouge et de développer un algorithme plus robuste aux projections et aux changements d'éclairage.

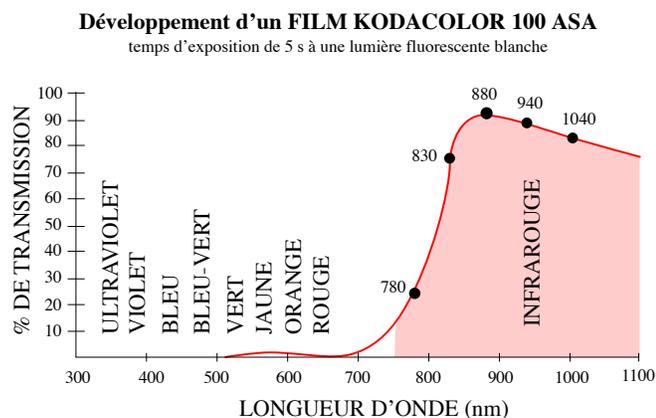


FIG. 4.2 – Filtre de lumière visible.

4.2 Modélisation des DLx

Dans la modélisation d'un DLx, quatre systèmes de coordonnées sont utilisés pour exprimer les poses relatives des composantes optiques. La caméra et le projecteur ont chacun un système de coordonnées, centré sur leur centre optique. On identifie ces systèmes en tant que *système de la caméra* et *système du projecteur*. La caméra et le projecteur se déplacent en fonction des rotations horizontale (*pan*) et verticale (*tilt*) de la tête robotisée. Pour modéliser ces déplacements, il faut pouvoir exprimer leur position en fonction du centre de rotation de la tête (figure 4.3). L'axe de rotation vertical et l'axe de rotation horizontal se croisent, ce qui limite le mouvement de la tête du DLx à un seul centre de rotation. Le troisième système de coordonnées se situe en ce centre de rotation, on y réfère par *système du DLx*. Finalement, on introduit le *système du monde* pour pouvoir exprimer la position du DLx (son centre de rotation) en fonction d'une référence absolue dans le monde. Ce dernier système de références permet d'exprimer la position de la caméra, du projecteur et des points 3D en fonction d'une référence absolue commune.

4.3 Calibration de la caméra

La calibration des paramètres internes et externes de la caméra des DLx permet de trouver la relation entre les points images et les points 3D d'une scène, même lorsque le DLx tourne. Pour exprimer les points 3D dans la référence du DLx, les paramètres

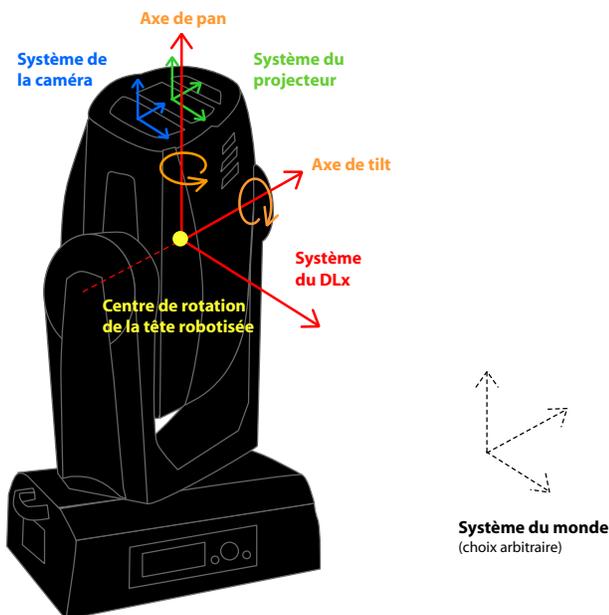


FIG. 4.3 – Modélisation d’un DLx.

externes de la caméra doivent être calibrés en fonction du centre de rotation de la tête robotisée du DLx. Selon les spécifications du fabricant, nous savons que la caméra des DLx a une résolution de 768×494 pixels.

4.3.1 Calibration des paramètres internes

Les caméras des DLx possèdent un zoom mécanique ($18\times$) et un zoom numérique ($12\times$). Ce dernier ne modifie pas la longueur focale de l’appareil, il ne fait qu’agrandir la dimension des pixels dans l’image. En observant les images et le déplacement de la lentille dans la caméra, nous estimons que le zoom numérique est actif à partir d’une valeur DMX d’environ 36000. Aucune calibration n’est faite au-delà puisque le zoom numérique n’apporte pas d’informations supplémentaires dans la capture. Chaque paramètre interne de la caméra est calibré sur 37 incréments du zoom mécanique.

Calibration de la distorsion radiale

Les images prises avec la caméra des DLx montrent une importante distorsion radiale. Avant de pouvoir calibrer les paramètres internes de la caméra, il est important de calibrer la distorsion radiale pour annuler son effet.

La technique de calibration utilisée est décrite précédemment aux sections 3.3 et 3.4.1. Nous tirons profit de la tête robotisée pour la collecte des données. Une ligne noire parfaitement droite contrastant avec le mur blanc nous sert d'objet de calibration. Pour chaque incrément de zoom, nous balayons la ligne en faisant tourner la caméra autour de l'axe vertical de façon à couvrir le champ de vision de la caméra et ainsi échantillonner l'effet de la distorsion radiale dans toute l'image (figure 4.4).

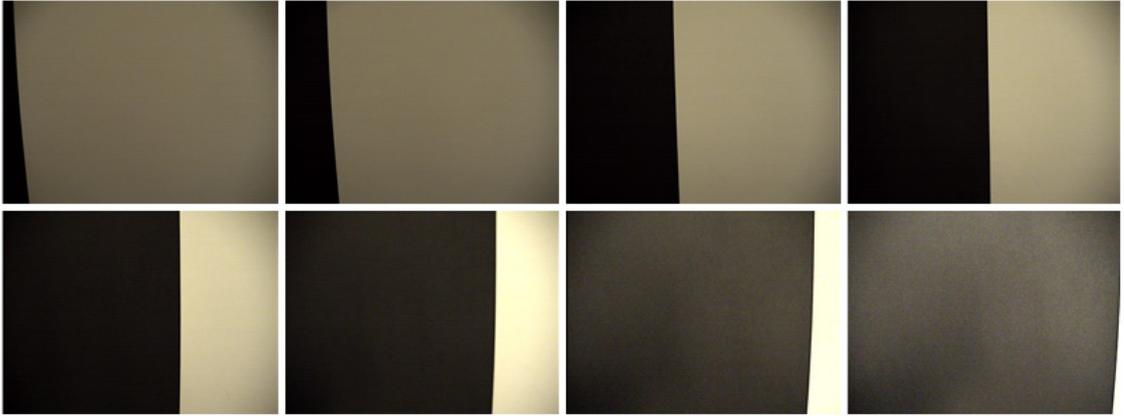


FIG. 4.4 – Échantillonnage de la distorsion radiale en observant une ligne. Les images permettent de remarquer le réglage automatique de la caméra et son vignette.

À l'aide d'un calcul de gradient horizontal, on extrait les coordonnées qui composent chaque ligne. En prenant des triplets de points au hasard sur chacune de ces lignes (équation 3.21), nous arrivons à déterminer les paramètres k_1 et k_2 de la fonction de distorsion radiale $d(r)$ pour les 37 incréments de zoom échantillonnés.

Il est difficile de calibrer tous les incréments de zoom. Pour modéliser la distorsion radiale de la caméra pour toutes les longueurs focales, nous interpolons les paramètres k_1 et k_2 en fonction de la valeur du zoom. La distribution de ces valeurs peut être représentée par une fonction de degré quatre en fonction de la valeur DMX du zoom

$$\begin{aligned} k_1(z) &= -0.289817 + 1.55806z - 3.12431z^2 + 2.75511z^3 - 0.899187z^4 \\ k_2(z) &= 0.7259653 - 0.415095z - 0.980476z^2 + 2.30989z^3 - 1.17813z^4 \end{aligned}$$

où z est la valeur normalisée du zoom DMX tel que $z = (\text{valeur DMX})/36000$. Les valeurs de k_1 et k_2 et leur interpolation sont illustrés à la figure 4.5. Ils déterminent la fonction du coefficient de distorsion radiale pour chaque valeur de zoom (figure 4.6).

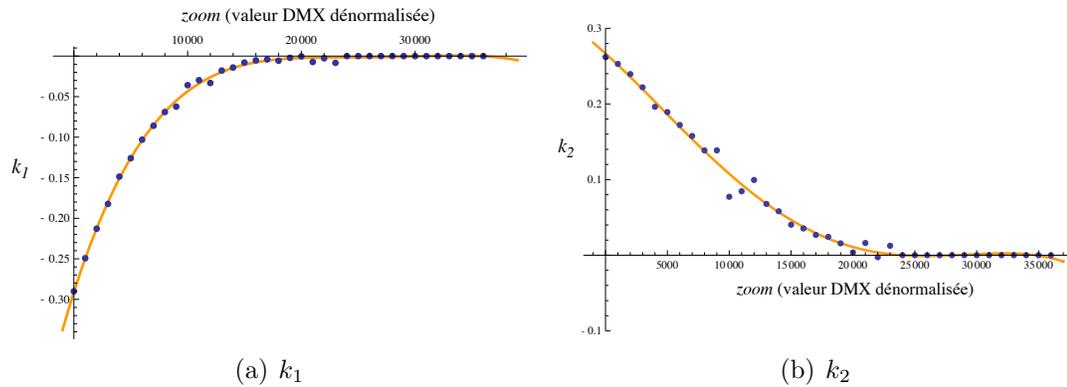


FIG. 4.5 – Valeurs discrètes de k_1 et k_2 trouvées lors de la calibration et leur interpolation respective en fonction de la valeur du zoom DMX.

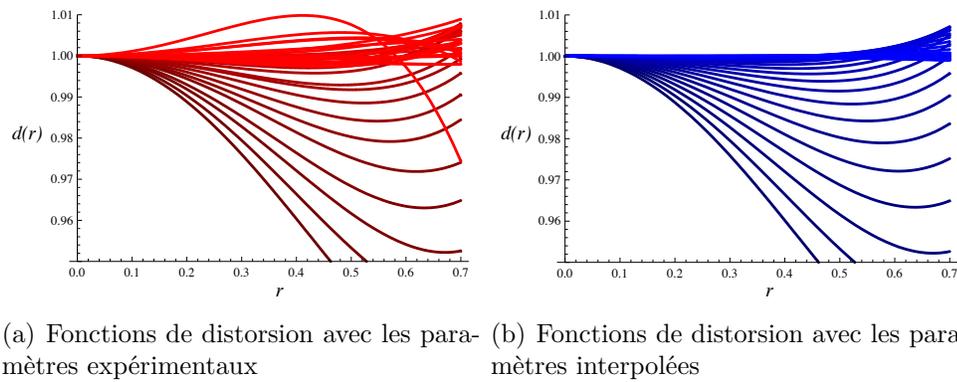


FIG. 4.6 – Fonctions du coefficient de distorsion radiale pour les 37 incréments de k_1 et k_2 calibrés.

Nous connaissons les valeurs de k_1 et k_2 pour chaque incrément de zoom, nous pouvons donc calculer la fonction de rectification de la distorsion radiale pour les 37 incréments de zoom calibrés. Grâce à l'interpolation des paramètres k_1 et k_2 , nous pouvons déterminer $d(r)$ pour tous les incréments de zoom. Nous pouvons donc exprimer la distorsion radiale de la caméra comme une fonction du rayon de distorsion

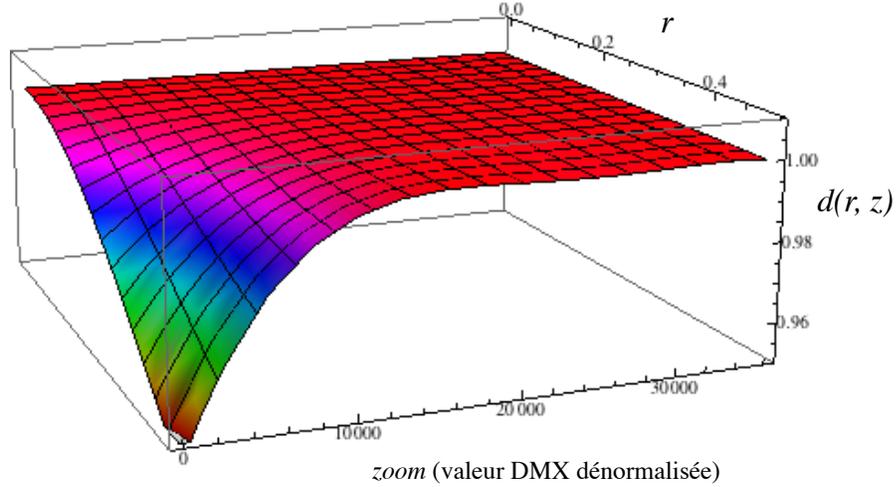


FIG. 4.7 – Fonction du coefficient de distorsion radiale $d(r, z)$ en fonction du zoom z et du rayon de distorsion r .

r et du zoom DMX z . La fonction $d(r, z)$ obtenue lors de la calibration d'un DLx est

$$d(r, z) = 1 + \begin{pmatrix} \mathbf{r}^2 & \mathbf{r}^4 \end{pmatrix} \begin{bmatrix} -0.288624 & 0.257558 \\ 1.55597 & -0.415067 \\ 3.12906 & -0.972493 \\ 2.75897 & 2.32976 \\ -0.896938 & -1.20566 \end{bmatrix}^T \begin{pmatrix} 1 \\ \mathbf{z} \\ \mathbf{z}^2 \\ \mathbf{z}^3 \\ \mathbf{z}^4 \end{pmatrix} \quad (4.1)$$

et est représentée à la figure 4.7.

Calibration de la longueur focale

Une fois la distorsion radiale de la caméra corrigée, la matrice de paramètres internes peut être calibrée sans être influencée par la déformation de l'image. Les paramètres internes de la caméra sont calculés à l'aide d'une calibration planaire. La modification du zoom entraîne un changement de longueur focale de la caméra. Pour calibrer entièrement les paramètres internes de la caméra, on échantillonne 37

incréments de zoom. En interpolant les paramètres internes expérimentaux, on peut représenter la longueur focale et le centre optique comme des fonctions dépendantes de la valeur DMX du zoom.

Pour l'échantillonnage des paramètres internes, une grille de calibration est créée spécialement pour permettre de couvrir la grande étendue de longueur focale de la caméra. Une petite grille incluse dans un des carrés de la grille principale assure la calibration des paramètres internes lorsque l'angle de vue de la caméra est trop faible et que la caméra ne peut pas voir la grille en entier (figure 4.8). En s'assurant que les deux grilles de calibration sont dans le même référentiel (le point $(0, 0, 0)$ est un des coins de la grande grille), les homographies trouvées permettent de retrouver \mathcal{K} pour chaque intervalle de zoom.

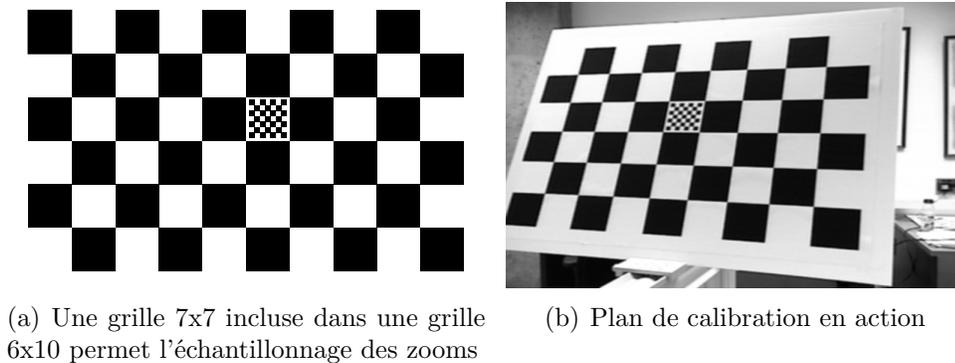


FIG. 4.8 – Patron de la grille de calibration.

Étant donné que le flux vidéo du DLx provient d'une caméra de mauvaise qualité et qu'il doit passer par un numériseur vidéo, on ne suppose pas que les pixels sont carrés ($f_x \neq f_y$). Les paramètres internes obtenus par calibration planaire pour un des DLx au zoom 0 sont $f_x = 861.903$, $f_y = 751.693$, $u = 394.372$ et $v = 233.453$.

L'interpolation des valeurs de la focale couvrant tout l'intervalle de zoom mécanique est présentée à la figure 4.3.1 et est obtenue par la fonction suivante :

$$f(z) = 0.0655813 + 0.340038z - 4.88739z^2 + 38.5508z^3 - 140.505z^4 \\ + 263.092z^5 - 242.735z^6 + 87.7801z^7$$

où z est la valeur DMX normalisée du zoom telle que $z = (\text{valeur DMX})/36000$ et $f(z)$ est une longueur focale normalisée telle que $f = f(z) \times 12000$.

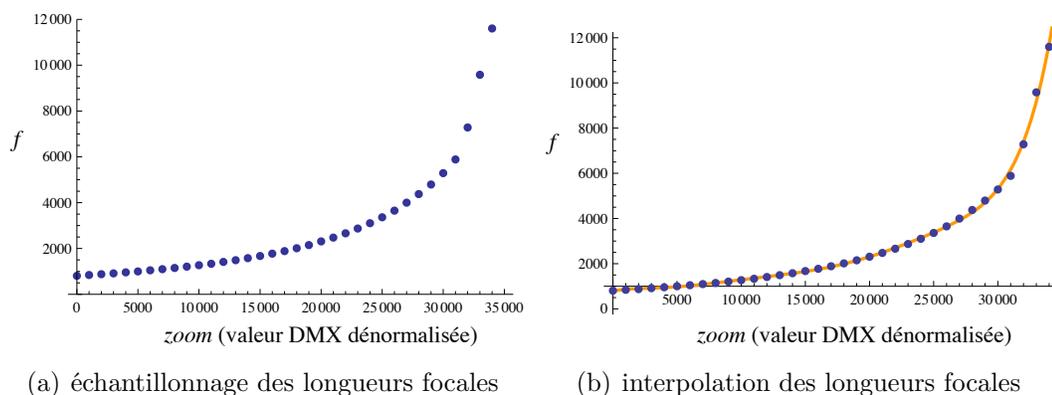


FIG. 4.9 – Calibration des focales en fonction des incréments de zoom d’une caméra DLx.

Calibration du centre optique

La caméra des DLx possède un agrandissement optique allant jusqu’à $18\times$. Étant donné l’intervalle important couvert par le zoom, on peut supposer que le déplacement du centre optique n’est pas négligeable. L’ajustement d’une courbe sur les échantillons calibrés nous permet d’interpoler la position du centre pour tous les incréments DMX du zoom de la caméra.

Cette observation est confirmée par les résultats de la calibration du centre optique avec la méthode de calibration planaire. Les résultats présentés à la figure 4.10(a) montrent un important déplacement vertical, alors que le déplacement horizontal est négligeable.

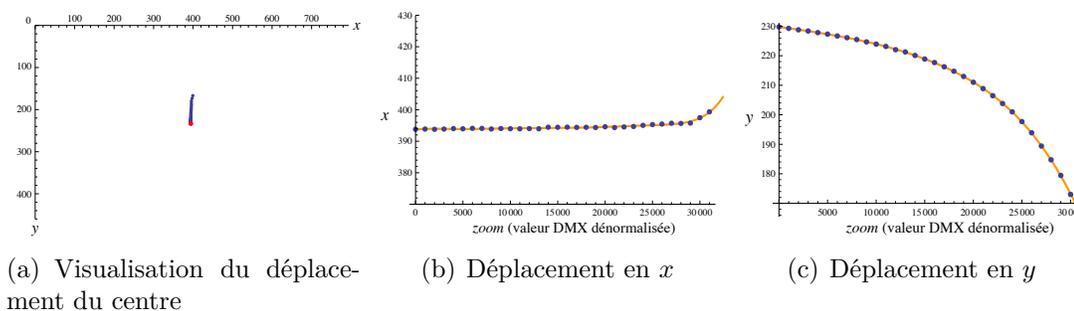


FIG. 4.10 – Calibration du point principal en fonction des incréments de zoom d’une caméra DLx.

La position y du centre optique de la caméra en fonction du zoom tel que présenté à la figure 4.3.1 est la suivante :

$$v(z) = 0.465084 - 0.0313419z - 0.0740318z^2 + 0.475997z^3 - 2.3555z^4 \\ + 4.90482z^5 - 4.99218z^6 + 1.83937z^7$$

où z est la valeur DMX normalisée du zoom telle que $z = (\text{valeur DMX})/36000$ et $v(z)$ est une coordonnée pixel en y normalisée telle que $y = v(z) \times 494$.

4.3.2 Calibration de la pose relative au monde du DLx

Les rotations horizontale et verticale du DLx permettent de couvrir un plus grand espace de reconstruction qu'une simple caméra. Pour exprimer les points 3D en fonction d'une référence fixe dans le monde, on doit estimer la position de la caméra de telle sorte qu'elle compense l'effet de la rotation de la tête du DLx. Étant donné que le centre de la caméra n'est pas positionné au centre de rotation de la tête robotisée, une rotation du DLx provoque une rotation et une translation de la position de la caméra. La calibration de la pose de la caméra correspond à déterminer la position de la caméra par rapport au centre de rotation du DLx. La rotation du DLx est connue puisqu'elle est contrôlée par DMX. En calibrant la pose de la caméra par rapport au centre de rotation, nous pouvons déduire la position de la caméra dans le monde en la combinant avec les mouvements du DLx.

Calibration par plan elliptique

La méthode de calibration décrite tire avantage des caractéristiques motrices des DLx (figure 4.11) :

1. les axes de rotation vertical et horizontal de la tête robotisée se croisent,
2. la tête du DLx peut être pivotée sur son axe horizontal pour s'aligner avec son axe de rotation vertical (nous l'appelons *la position 0*),
3. le DLx tourne autour de son axe vertical sur 400°.

Quelques hypothèses sur le modèle du DLx nous permettent de faciliter la calibration (figure 4.11) :

1. le centre du DLx se trouve au croisement de ses deux axes de rotation,
2. en position 0, l'axe de la caméra et l'axe du DLx ont la même orientation,
3. le projecteur est centré en x avec le système d'axe du DLx.

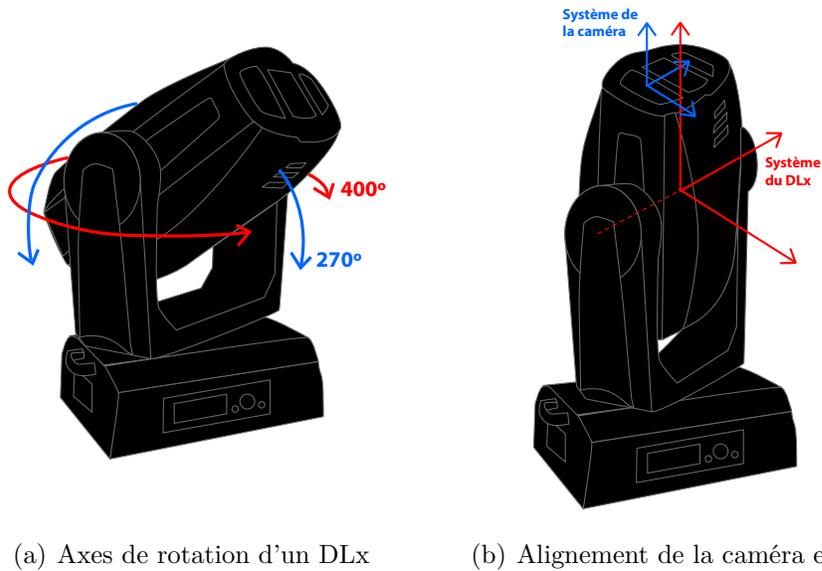


FIG. 4.11 – Modélisation du DLx.

Ces caractéristiques nous permettent de générer un plan elliptique en observant un seul point visible par la caméra alors que le DLx tourne autour de son axe vertical (figure 4.12). En position 0, l'accumulation des coordonnées du point visible sur plusieurs incréments de rotation verticale donnera un cercle. En calculant la transformation projective 2D (\mathcal{H}_0) qui permet de projeter les coordonnées du plan circulaire sur le plan de la caméra, nous pouvons déterminer la transformation $\mathcal{P}_{plan} = [\mathcal{R}_{plan} | \mathbf{t}_{plan}]$ reliant la caméra en position 0 et le plan circulaire de calibration. Nous pouvons

extraire une rotation et une translation de l'homographie par

$$\begin{aligned}\mathcal{H}_0 &= \mathcal{K}\mathcal{P}_{plan} \\ \mathcal{K}^{-1}\mathcal{H}_0 &= \mathcal{P}_{plan} \\ \mathcal{K}^{-1}\mathcal{H}_0 &= [\mathcal{R}_{plan} | \mathbf{t}_{plan}] \\ \mathcal{K}^{-1}\mathcal{H}_0 &= \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}_{plan} \end{bmatrix}\end{aligned}$$

où \mathbf{r}_1 et \mathbf{r}_2 sont les colonnes 1 et 2 de la matrice de rotation \mathcal{R}_{plan} . Pour construire la matrice complète \mathcal{R}_{plan}

$$\mathcal{R}_{plan} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_1 \times \mathbf{r}_2 \end{bmatrix} \quad (4.2)$$

En position 0, la rotation \mathcal{R}_{cam} sera très près de l'identité puisque les axes de la caméra et du DLx ont la même orientation, mais une légère rotation \mathcal{R}_{cam} est probable étant donné le montage manuel du boîtier. Quant à la translation \mathbf{t}_{plan} trouvée grâce à \mathcal{H}_0 et à \mathcal{K} , elle représente la translation entre le centre de la caméra et le centre du plan circulaire. Une fois ces transformations trouvées, il reste à déterminer la translation \mathbf{t}_{cam} qui relie la caméra au centre de rotation du DLx et à exprimer la pose de la caméra dans le système du DLx.

En ajoutant une légère rotation horizontale au DLx en position 0, les coordonnées accumulées génèrent une ellipse qui est reliée au cercle par une homographie. La rotation et la translation de la caméra extraites de l'homographie permettent de positionner la caméra par rapport à la position 0. En accumulant des plans elliptiques sur plusieurs valeurs de rotations horizontales, on peut retrouver la translation \mathbf{t}_{cam} entre le centre de rotation horizontal du DLx et le centre de caméra. On détermine le centre de rotation du DLx en intersectant les axes optiques des caméras trouvés sur chaque incrément de rotation horizontale. Un exemple de calibration est illustré à la figure 4.13 de la page 42. Le point obtenu est une première approximation de la translation \mathbf{t}_{dlx} entre le centre de rotation du DLx et le plan de calibration.

On obtient la pose de la caméra dans le système du DLx par

$$\begin{aligned}\mathcal{P}_{cam} &= [\mathcal{R}_{plan} | \mathbf{t}_{plan} - \mathbf{t}_{dlx}] \\ \mathcal{P}_{cam} &= [\mathcal{R}_{plan} | \mathbf{t}_{cam}].\end{aligned}$$

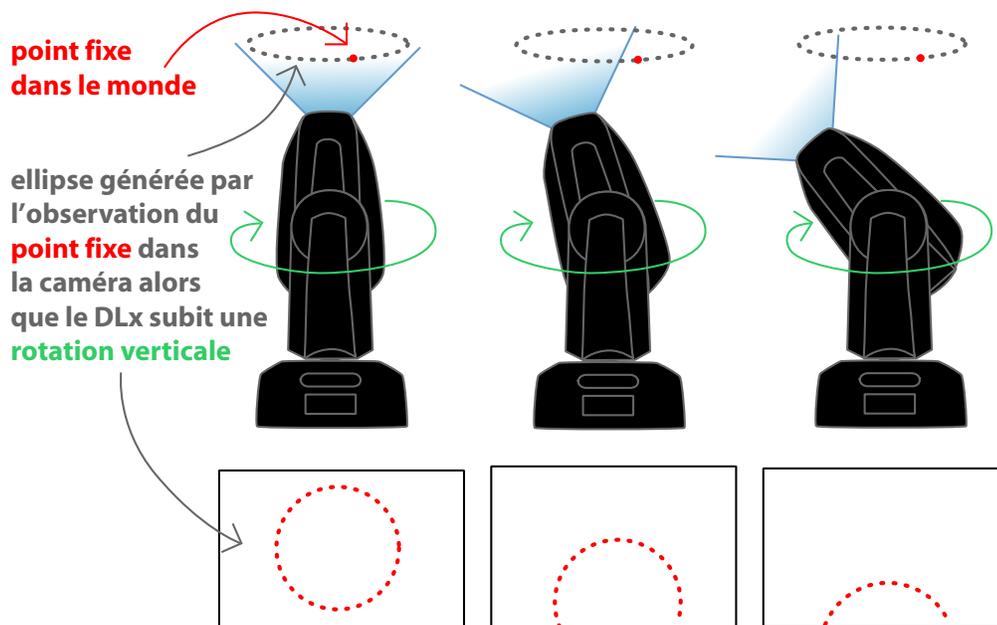


FIG. 4.12 – Génération de plans elliptiques par la rotation de la caméra autour de l'axe vertical en regardant un point dans le monde (en rouge).

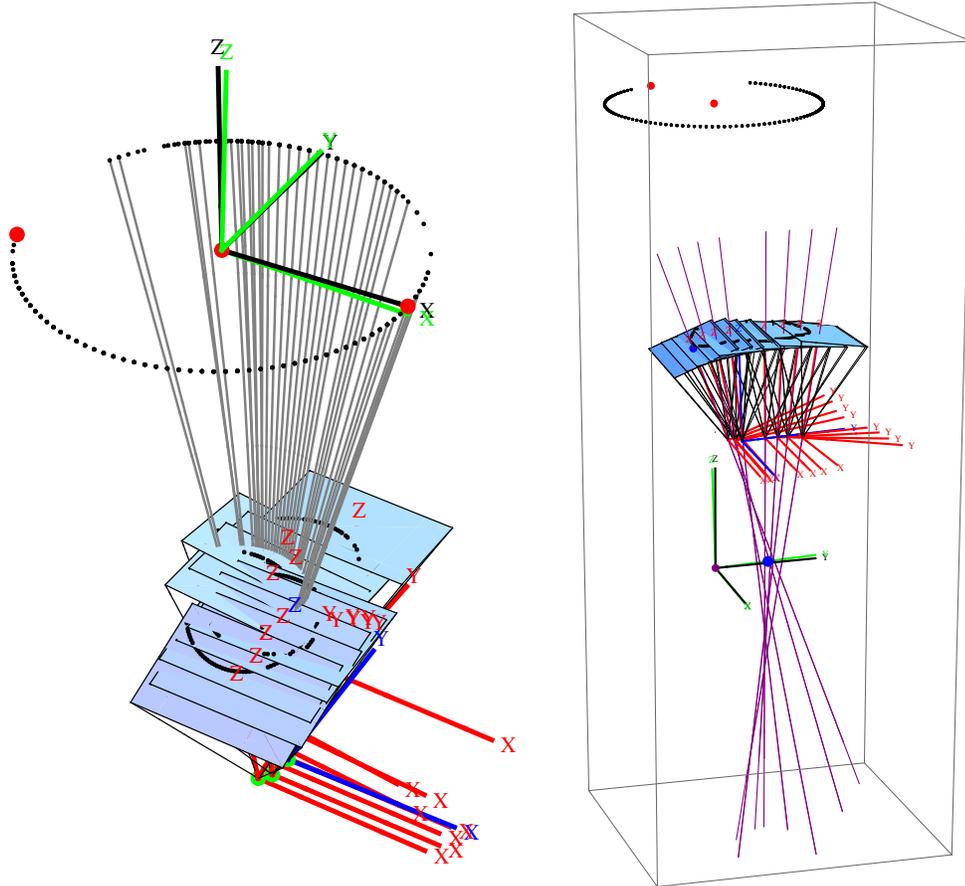
L'unité de mesure de la pose actuelle est le rayon du cercle de calibration. Pour exprimer la position de la caméra selon le système métrique, il faut déterminer la valeur métrique du rayon du cercle de calibration. Pour ce faire, on utilise un objet de calibration de longueur connue, placé à la même distance de la caméra que le point qui a servi à générer les plans elliptiques. En mesurant la longueur de l'objet dans l'image, on obtient le rapport pixels/cm. En ajustant la courbe quadratique d'une ellipse ($ax^2 + 2bxy + cy^2 + 2dx + 2fy + g = 0$) sur les coordonnées images du plan elliptique de calibration et en trouvant le centre (x_0, y_0) où $x_0 = (cd - bf)/b^2 - ac$ et $y_0 = (af - bd)/b^2 - ac$, on détermine la valeur du rayon en pixels dans l'image. Le rapport pixels/cm nous permet d'exprimer le rayon du cercle de calibration en centimètres, et d'exprimer la pose de la caméra par rapport au DLx en centimètres.

Nous connaissons l'angle horizontal de la tête robotisée puisqu'il est contrôlé par DMX et nous connaissons la position de la caméra par rapport à l'axe de rotation. Il est donc possible de calculer la position de la caméra dans le système du DLx malgré

le mouvement de la tête robotisée avec

$$\mathcal{P}_{cam} \mathcal{R}_{tilt} \mathcal{R}_{pan} \quad (4.3)$$

où \mathcal{R}_{tilt} est la rotation de la tête robotisée autour de l'axe horizontal et \mathcal{R}_{pan} est la rotation autour de l'axe vertical. Nous pouvons déterminer les matrices de rotation \mathcal{R}_{tilt} et \mathcal{R}_{pan} facilement puisque nous savons que les valeurs DMX de rotations horizontale et verticale, de 0 à 65535, sont distribuées linéairement et couvrent respectivement 270° et 400° .



(a) Caméras relatives au plan de calibration (b) Caméras relatives au centre du DLx

FIG. 4.13 – Retrouver le centre du DLx à partir des poses ayant subies une rotation horizontale.

Il serait possible d'estimer la pose de la caméra à partir d'homographies issues d'un simple plan rectangulaire placé perpendiculairement à l'axe vertical. Par contre, un plan rectangulaire restreint l'angle maximum de rotation verticale puisque la caméra doit voir quatre points en tout temps. L'observation d'un seul point pendant la rotation verticale de la tête robotisée augmente l'angle maximum de rotation horizontale puisque nous pouvons retrouver le plan elliptique même si seulement une section est générée. La plus grande amplitude de rotation obtenue par l'utilisation de plans elliptiques permet de retrouver le centre de rotation avec une plus grande précision, puisque les axes optiques se croiseront avec un plus grand angle, d'où l'intérêt de la technique de calibration proposée.

Ajustement de la pose

Les poses de caméra trouvées à l'aide des plans elliptiques ne permettent pas d'isoler un centre de rotation commun clairement défini (figure 4.13). Le centre de rotation trouvé est une première estimation. Étant donné que la pose de caméra obtenue sert de référence à la pose du projecteur, il est important de raffiner le résultat pour éviter que l'erreur ne se propage dans les étapes de calibration suivantes. L'ajustement de la pose de caméra passe par une minimisation non-linéaire de l'erreur de reprojection du cercle de calibration dans la caméra. On minimise l'erreur de reprojection des points dans la caméra en ajustant la rotation de la caméra en position 0 par rapport au plan de calibration (\mathcal{R}_{plan}), la position de la caméra dans le système du DLx (\mathbf{t}_{cam}) et la distance entre le centre de rotation du DLx et le plan elliptique de calibration (z_{plan}) trouvée en intersectant les axes optiques des caméras calibrées autour de l'axe horizontal. Puisqu'on estime que le DLx est perpendiculaire au plan de calibration en position 0, l'intégration de \mathcal{R}_{plan} dans la minimisation permet d'estimer le dé-alignement de la caméra par rapport à l'axe du DLx.

L'ajustement de pose consiste à minimiser les paramètres de rotation $\theta_x, \theta_y, \theta_z$, les paramètres de translation t_x, t_y, t_z et z_{plan} qui composent

$$\begin{aligned}\mathcal{R}_{plan} &= \mathcal{R}_{\theta_x} \mathcal{R}_{\theta_y} \mathcal{R}_{\theta_z} \\ \mathbf{t}_{cam} &= \begin{pmatrix} t_x & 0 & t_z \end{pmatrix}^T \\ \mathbf{P}_{plan[t]}^i &= \begin{pmatrix} x_{plan[t]}^i & y_{plan[t]}^i & z_{plan} \end{pmatrix}^T\end{aligned}$$

dans

$$\sum_t^{nbtilt} \sum_i^{nbpts} D(\mathbf{p}_{cam[t]}^i, \mathcal{K}[\mathcal{R}_{plan} | \mathbf{t}_{cam}] \mathcal{R}_{tilt[t]} \mathbf{P}_{plan[t]}^i) \quad (4.4)$$

où $D(u, v)$ est la distance euclidienne entre deux points projectifs u et v , $tilt[t]$ est la valeur de la rotation horizontale en position t et $\mathbf{p}_{cam[t]}$ sont les points images associés aux points du plan de calibration $\mathbf{P}_{plan[t]}$ en position t . On suppose $\mathbf{t}_{cam} = (t_x \ 0 \ t_z)^T$ puisque par observation, la caméra est centrée sur l'axe de rotation horizontal en y . La nouvelle valeur de z_{plan} servira à calculer la nouvelle valeur métrique entre le centre de caméra et le centre de rotation du DLx. Le résultat de l'ajustement de la calibration est illustré à la figure 4.14 de la page 45.

Le résultat de la calibration par plans elliptiques donne une translation $\mathbf{t}_{cam} = (0,38 \ 0,15 \ 1,37)^T$ en rayon de cercle. Sachant que le rayon du plan de calibration était de 28,95cm, on obtient une translation initiale de

$$\mathbf{t}_{cam} = \left(11,14 \ 4,42 \ 39,74 \right)^T$$

qui devient

$$\mathbf{t}_{cam} = \left(9,68 \ 0,00 \ 24,40 \right)^T$$

après la minimisation.

4.4 Calibration du projecteur

La calibration du projecteur est essentielle pour projeter des vidéos adaptatifs sur les faces des cubes en mouvement. Une fois les paramètres internes estimés, il est possible de retrouver la pose du projecteur par rapport à la caméra en utilisant la géométrie épipolaire. Puisque la pose de la caméra dans le système du DLx est connue, la géométrie épipolaire permettra de trouver la pose du projecteur par rapport au même système du DLx.

4.4.1 Estimation des paramètres internes

Les paramètres internes du projecteur sont estimés expérimentalement en calculant l'angle de vue du projecteur. Bien que le projecteur soit muni d'un zoom optique de 1,3×, seul les paramètres au zoom le plus large ont été calibrés puisque cela était

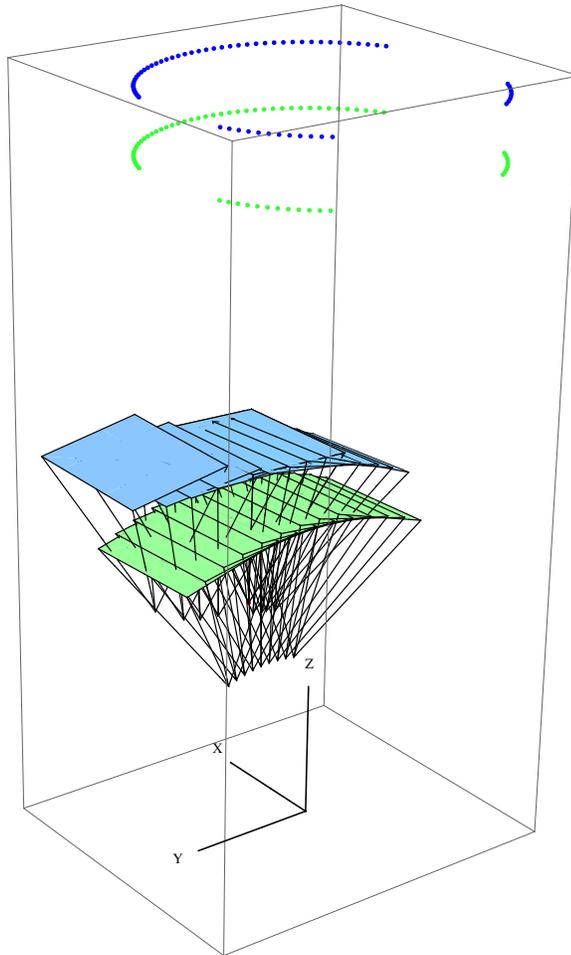


FIG. 4.14 – Résultat de la minimisation de la pose de caméra dans le système du DLx.

suffisant pour nos applications. Un écran placé à 236 cm du projecteur nous permet d'établir le ratio de projection et de calculer la longueur focale :

$$f_x = \frac{r_x}{W/D} = \frac{1024 \text{ px}}{102 \text{ cm}/236 \text{ cm}} = 2369,25$$

$$f_y = \frac{r_y}{H/D} = \frac{768 \text{ px}}{76 \text{ cm}/236 \text{ cm}} = 2384,84$$

$$f = \frac{f_x + f_y}{2} = \frac{2369,25 + 2384,84}{2} \approx 2377$$

où (r_x, r_y) est la résolution du projecteur, W est la largeur de la projection à une distance D donnée, et H est la hauteur de la projection à la même distance.

Dans un projecteur, le centre optique est généralement situé en bas au centre de l'image pour permettre de projeter des images pleine hauteur lorsque le projecteur est placé à plat sur une surface. Avant de tirer cette conclusion avec le projecteur des DLx, nous vérifions le centre de l'image expérimentalement en utilisant le zoom optique du projecteur. En plaçant un plan face au projecteur et en variant la valeur du zoom, nous remarquons que le changement d'échelle de la projection est fixe au centre de la projection. Nous estimons donc le centre optique du projecteur en plein centre de la projection, $(c_x, c_y) = (512, 384)$.

La matrice de paramètres internes du projecteur est estimée à

$$\mathcal{K}_{proj} = \begin{bmatrix} 2377 & 0 & 512 \\ 0 & 2377 & 384 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.5)$$

Il serait possible de faire cette calibration automatiquement plutôt que de l'estimer, mais nous croyons que la précision gagnée serait faible. Par contre, la calibration manuelle est facilement invalidée puisque les contrôles des DLx permettent à l'utilisateur de déplacer l'objectif des projecteurs (par exemple, un contrôle du *lens shift*). Dans le cas spécifique des DLx, une procédure de calibration automatique serait bénéfique puisqu'elle s'adapterait facilement à ce genre de changement. Pour éviter des problèmes de calibration, nous avons simplement évité de modifier les paramètres des projecteurs pendant le déploiement de nos applications avec les DLx.

4.4.2 Calibration de la pose dans le système du DLx

La géométrie épipolaire nous permet de mettre les vues de la caméra et du projecteur en correspondance et de calibrer la pose du projecteur. Les correspondances entre le projecteur et la caméra sont générées par lumière structurée.

Création des correspondances par lumière structurée

La lumière structurée est une technique de reconstruction 3D basée sur une série d'images de patrons binaires projetées et vues par une caméra [38]. Chaque coordon-

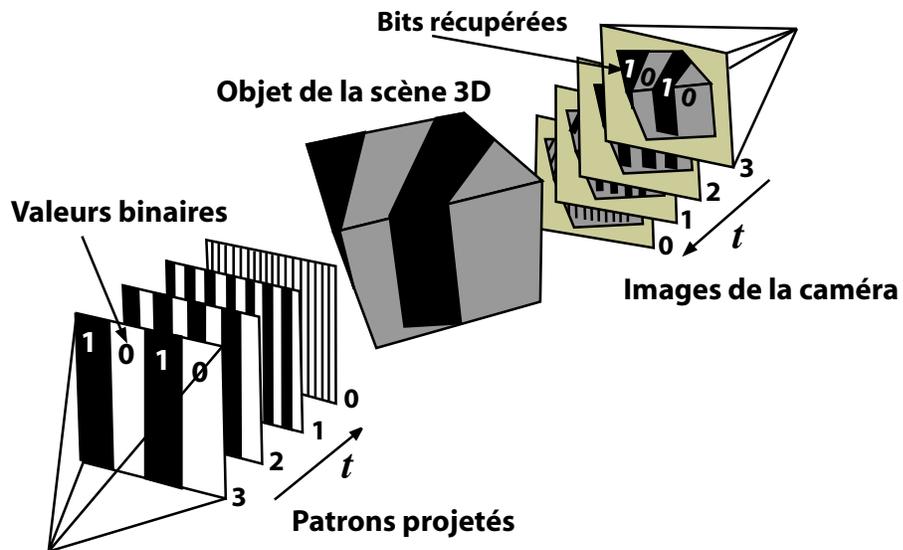


FIG. 4.15 – Exemple de lumière structurée pour les bits 0 à 3 en x .

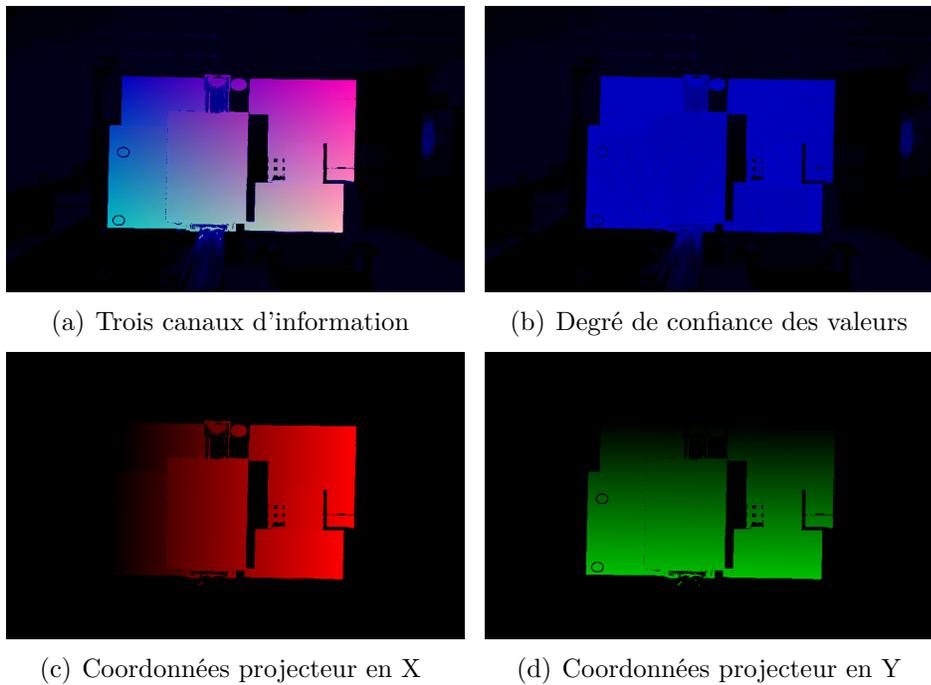


FIG. 4.16 – Résultat de la lumière structurée sauvegardé dans une image.

née pixel du projecteur est représentée par une valeur binaire. L'accumulation des patrons de lumière structurée dans le temps formera un code binaire. En observant la projection de ces patrons avec une caméra il est possible de récupérer les codes binaires projetés. Le code récupéré dans un pixel donné de la caméra correspond à une coordonnée dans le projecteur, ce qui nous donne directement une correspondance caméra-projecteur. Cette technique de mise en correspondance est illustrée à la figure 4.15.

Étant donné que la calibration de la pose du projecteur sera déterminée à partir des contraintes de la géométrie épipolaire, il est important que les correspondances de points couvrent un grand intervalle de profondeur pour augmenter la stabilité numérique de la solution de pose. Pour ce faire, la lumière structurée a été projetée sur une scène profonde contenant plusieurs plans à différentes profondeurs (figure 4.16). La lumière structurée génère un très grand nombre de correspondances mais un échantillonnage réduit suffit pour la calibration par géométrie épipolaire.

Calcul de la pose

À partir des correspondances caméra-projecteur obtenues avec la lumière structurée (figure 4.17), nous retrouvons la matrice fondamentale telle que décrite à la section 3.2. Les paramètres internes étant déjà estimés, nous calculons la matrice essentielle à partir de laquelle nous pouvons extraire les quatre poses possibles. Étant donné les connaissances a priori du DLx, nous savons que le projecteur est positionné à la gauche de la caméra et qu'une correspondance de points doit être reconstruite en z positif (devant le projecteur), ce qui nous permet de choisir facilement la bonne pose.

La pose du projecteur obtenue à l'aide de la matrice essentielle est relative à la caméra. Pour exprimer la pose du projecteur dans le monde du DLx

$$\mathcal{P}_{proj} = \mathcal{P}_{cam-proj} \mathcal{P}_{cam} \quad (4.6)$$

où $\mathcal{P}_{cam-proj}$ est la pose du projecteur relative à la caméra obtenue avec la matrice essentielle.

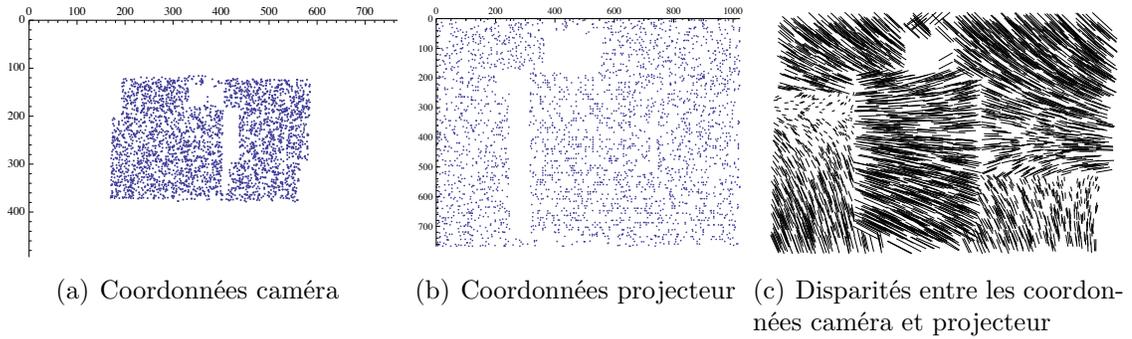


FIG. 4.17 – Correspondances extraites de la lumière structurée.

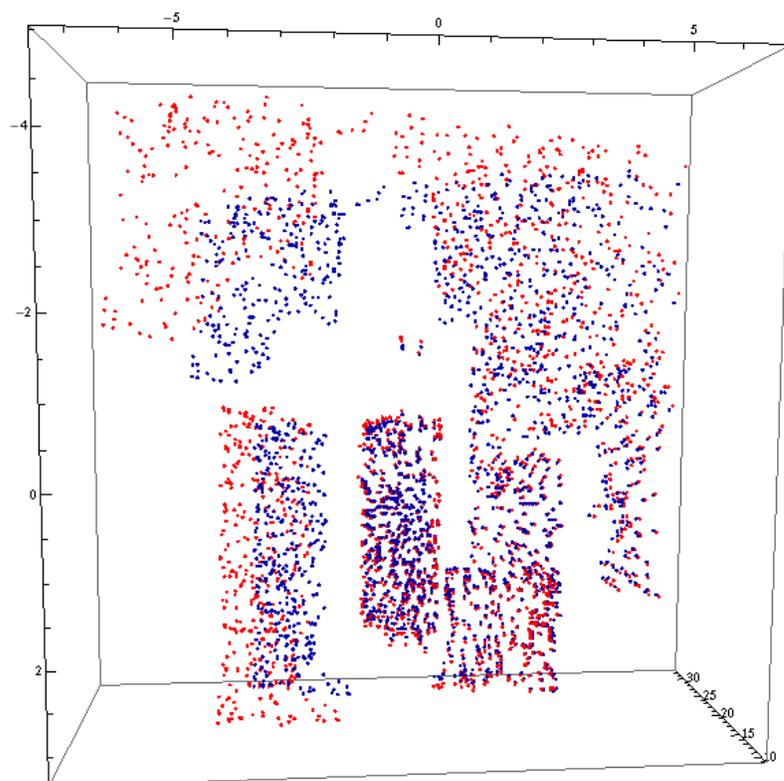
4.5 Correction de la calibration caméra-projecteur par la méthode de minimisation non-linéaire de Levenberg-Marquardt

Une calibration obtenue par la géométrie épipolaire n'est pas précise. Pour augmenter la fiabilité de la calibration, une étape de raffinement s'impose. Nous connaissons tous les paramètres nécessaires à la modélisation du DLx : les paramètres internes ($d(r)$, \mathcal{K}_{cam} et \mathcal{K}_{proj}), les poses par rapport au DLx (\mathcal{P}_{cam} et \mathcal{P}_{proj}), la pose du projecteur relative à la caméra ($\mathcal{P}_{cam-proj}$) et les matrices de rotation du DLx (\mathcal{R}_{pan} et \mathcal{R}_{tilt}). Étant donné que les paramètres internes des projecteurs sont estimés manuellement, il est probable qu'une erreur importante se glisse dans la calibration. Pour diminuer l'impact de l'erreur introduite par l'estimation des paramètres, les paramètres internes et externes du projecteur sont ajustés pour minimiser l'erreur de reprojection des points 3D dans le projecteur.

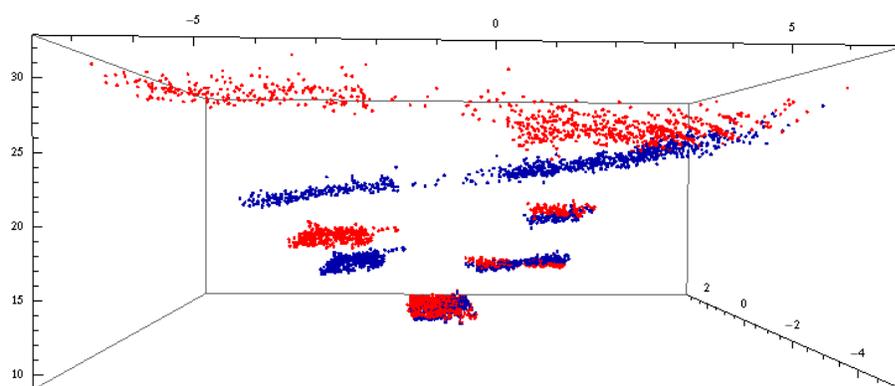
Puisque la minimisation de l'erreur de reprojection ne tient compte que du projecteur, on suppose $\mathcal{P}_{cam} = \mathbf{I}$, ce qui revient à minimiser la pose relative du projecteur par rapport à la caméra. Les paramètres relatifs de rotation θ_x , θ_y , θ_z , les paramètres relatifs de translation t_x , t_y , t_z , les paramètres internes f, u, v , et la profondeur z^i des points 3D sont ajustés pour minimiser l'équation

$$\min \sum_i^{\text{nb pixel}} D(\mathbf{p}_{proj}^i, \mathcal{K}_{proj}[\mathcal{R}_{cam-proj} | \mathbf{t}_{cam-proj}] \mathbf{P}^i) \quad (4.7)$$

où D est la distance euclidienne, \mathbf{p}_{proj}^i est un point 2D de référence dans l'image du projecteur et \mathbf{P}^i est un point 3D obtenu à partir du point \mathbf{p}_{cam}^i de l'image caméra,



(a) Vue de face



(b) La vue de haut montre que la reconstruction de la minimisation est beaucoup plus près de la réalité puisque le DLx était placé face aux plans

FIG. 4.18 – Résultat de la minimisation Levenberg-Marquardt dans la reconstruction des points 3D. Les points originaux sont en bleu et les points résultants de la minimisation sont en rouge.

projeté à une profondeur z^i et transformé en coordonnée du monde (\mathbf{p}_{cam}^i et \mathbf{p}_{proj}^i sont des correspondances 2D obtenues par lumière structurée). \mathbf{P}^i est donc défini à partir de $\mathbf{p}_{cam}^i = (x_{cam}^i, y_{cam}^i)$ tel que

$$\mathbf{P}^i = \mathcal{M}_{cam}^{-1} \begin{bmatrix} x_{cam}^i & y_{cam}^i & 1 & 1/z^i \end{bmatrix}^\top \quad (4.8)$$

où z^i appartient à la triangulation des points caméra (\mathbf{p}_{cam}^i) et projecteur (\mathbf{p}_{proj}^i) avec les matrices \mathcal{M}_{cam} et \mathcal{M}_{proj} qui sont issues de l'équation 2.11 avec

$$\mathbf{K}_{proj} = \begin{bmatrix} f & 0 & u \\ 0 & f & v \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathcal{R}_{cam-proj} = \mathcal{R}_{\theta_x} \mathcal{R}_{\theta_y} \mathcal{R}_{\theta_z}$$

$$\mathbf{t}_{cam-proj} = \begin{pmatrix} t_x & t_y & t_z \end{pmatrix}^\top.$$

Cette étape de minimisation non-linéaire ajuste la calibration du projecteur plus précisément et permet une meilleure mise en correspondance entre la caméra et le projecteur, ce qui résulte en une meilleure reconstruction 3D (figure 4.18). La distribution des erreurs de reprojection dans la caméra et le projecteur avant et après la minimisation (figure 4.19) montre l'importance de l'ajustement effectué. L'étape de minimisation utilise l'algorithme de Levenberg-Marquardt, une descente de gradient basée sur la méthode des moindres carrés. Dans le cas présent, le minimum de la fonction de coût correspond aux paramètres externes et internes du projecteur qui expliquent le mieux les disparités entre les correspondances de coordonnées caméra-projecteur calculées par lumière structurée.

La matrice de paramètres internes du projecteur après la minisation des paramètres internes et externes devient

$$\mathcal{K}_{proj} = \begin{bmatrix} 2367.07 & 0 & 496.43 \\ 0 & 2367.07 & 334.37 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.9)$$

contrairement aux valeurs estimées initiales, $f = 2377$, $u = 512$ et $v = 384$. La pose

relative du projecteur avait comme valeurs initiales

$$\begin{aligned}\mathcal{R}_{cam-proj} &= \mathcal{R}_x(0.72^\circ) \cdot \mathcal{R}_y(-1, 25^\circ) \cdot \mathcal{R}_z(1, 14^\circ) \\ \mathbf{t}_{cam-proj} &= \begin{pmatrix} -12.13 & -0.38 & -10.52 \end{pmatrix}^\top \text{ cm}\end{aligned}$$

et est passé à

$$\begin{aligned}\mathcal{R}_{cam-proj} &= \mathcal{R}_x(0.78^\circ) \cdot \mathcal{R}_y(0, 06^\circ) \cdot \mathcal{R}_z(-1, 17^\circ) \\ \mathbf{t}_{cam-proj} &= \begin{pmatrix} -10.94 & -0.23 & -9.43 \end{pmatrix}^\top \text{ cm}\end{aligned}$$

après la minimisation. Il ne reste qu'à exprimer la pose du projecteur par rapport au DLx (équation 4.6) pour compléter la calibration du DLx.

Il est maintenant possible de modéliser la position de la caméra et du projecteur peu importe la rotation de la tête robotisée. On exprime les poses de la caméra et du projecteur selon une référence absolue dans le monde par

$$\begin{aligned}\mathcal{P}_{cam}^{monde} &= \mathcal{P}_{cam} \mathcal{R}_{tilt} \mathcal{R}_{pan} \mathcal{P}_{dlx} \\ \mathcal{M}_{cam}^{monde} &= \mathcal{K}_{cam} \mathcal{P}_{cam}^{monde} \\ \mathcal{P}_{proj}^{monde} &= \mathcal{P}_{proj} \mathcal{R}_{tilt} \mathcal{R}_{pan} \mathcal{P}_{dlx} \\ \mathcal{M}_{proj}^{monde} &= \mathcal{K}_{proj} \mathcal{P}_{proj}^{monde}\end{aligned}$$

où \mathcal{R}_{tilt} et \mathcal{R}_{pan} sont les rotations verticale et horizontale de la tête commandée par DMX, et \mathcal{P}_{dlx} est la pose du DLx selon une référence du monde (si le monde n'est pas centré sur le DLx). Avec le modèle complet, il est maintenant possible de suivre un objet avec la tête robotisée tout en sachant sa position 3D absolue malgré les rotations du DLx.

4.6 Résultats

Le processus de calibration est composé de plusieurs étapes de calibration qui modélisent chacune des composantes optiques de façon indépendante. Chaque étape de calibration a été optimisée pour minimiser l'erreur de reprojection des points

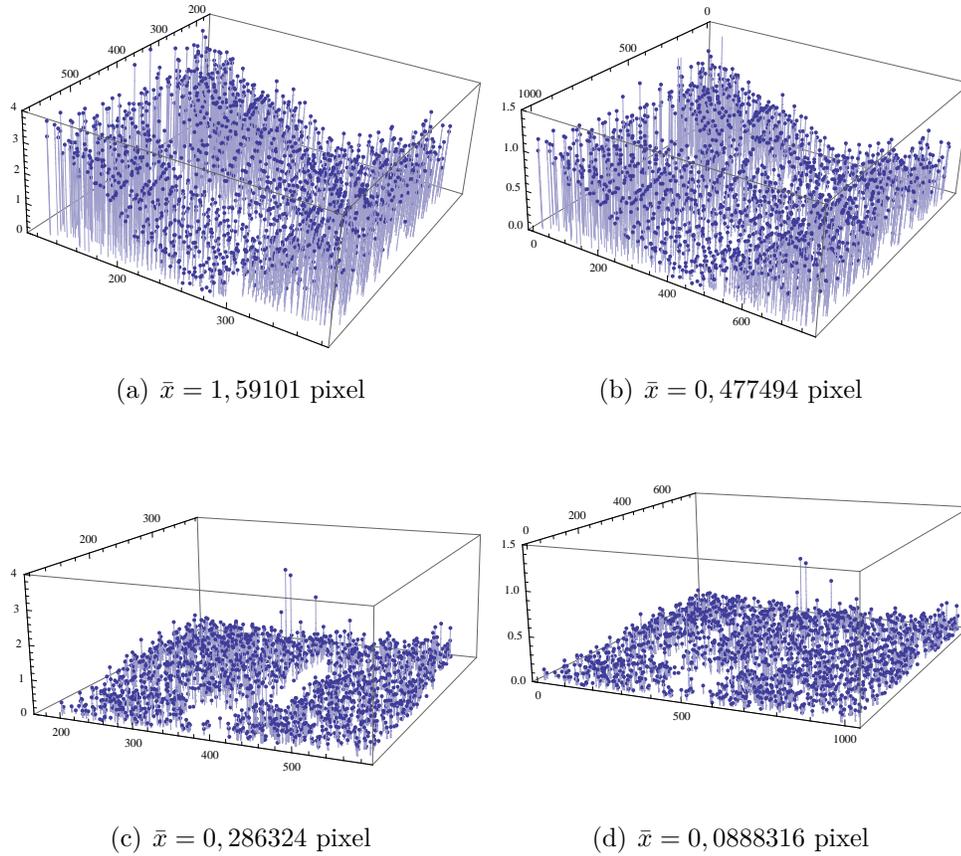


FIG. 4.19 – Erreurs de reprojection des points 3D dans la caméra avant (a) et après (c) la minimisation. Erreurs de reprojection des points 3D dans le projecteur avant (b) et après (d) 20 itérations de minimisation Levenberg-Marquadt.

dans les modèles de caméra. Pour vérifier la qualité de la calibration, toutes ces composantes doivent être mises en relation et testées dans une référence commune, soit le monde du DLx.

4.6.1 Calibration caméra-projecteur

La relation caméra-projecteur est calibrée par géométrie épipolaire, aux sections 4.4 et 4.5, en utilisant des correspondances obtenues par lumière structurée. La qualité de la mise en correspondance caméra-projecteur dépend de la calibration des paramètres internes de la caméra et du projecteur ainsi que de la pose relative du projecteur par rapport à la caméra.

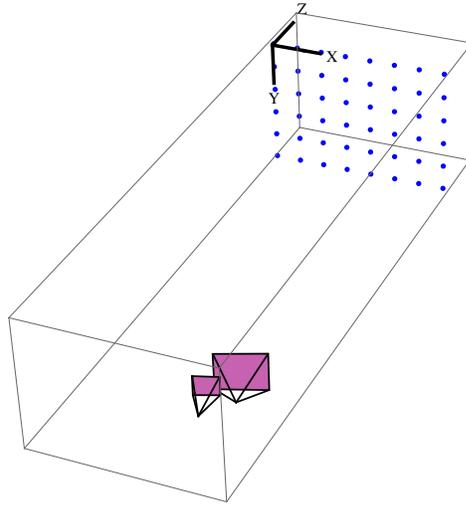


FIG. 4.20 – Visualisation des poses caméra et projecteur et reconstruction 3D du plan de calibration.

Pour vérifier la qualité de la calibration caméra-projecteur, on s'assure que la projection d'un objet 3D avec le projecteur correspond à l'image de l'objet observé par la caméra. Pour ce faire, un patron d'échiquier est placé devant la caméra. Les coordonnées 3D du plan reconnu par la caméra définissent une homographie par laquelle nous pouvons extraire la pose de la caméra par rapport au plan qu'on définit en tant qu'origine du monde (figure 4.20). Grâce à la calibration caméra-projecteur, nous déduisons la pose du projecteur par rapport au plan et reprojets les coordonnées 3D du plan dans l'image du projecteur. La projection des pixels projecteurs devrait correspondre parfaitement avec le plan observé.

Nos résultats montrent une erreur d'alignement entre 0,1 cm et 0,7 cm sur un plan situé à 2 m de la caméra (figure 4.21).

4.6.2 Calibration du modèle complet de DLx

La section précédente n'évaluait que la relation entre la caméra et le projecteur. Pour évaluer la qualité de la calibration complète du DLx, nous devons ajouter la rotation de la tête robotisée dans les variables à tester.

Le test élaboré pour évaluer la précision de la calibration consiste à choisir un point 3D dans le monde et à comparer la prédiction de sa projection dans les images de la caméra et du projecteur avec l'observation de sa projection dans ces images, et

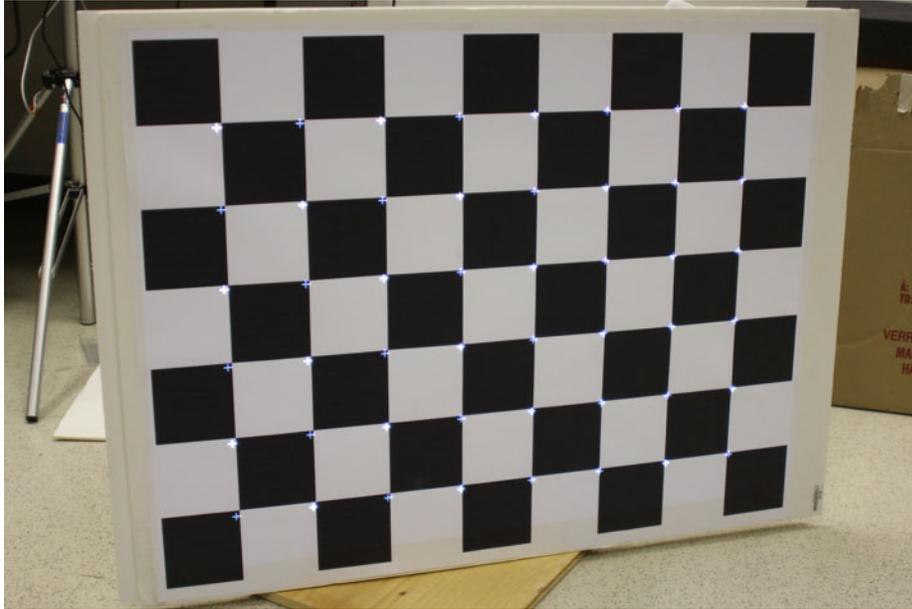
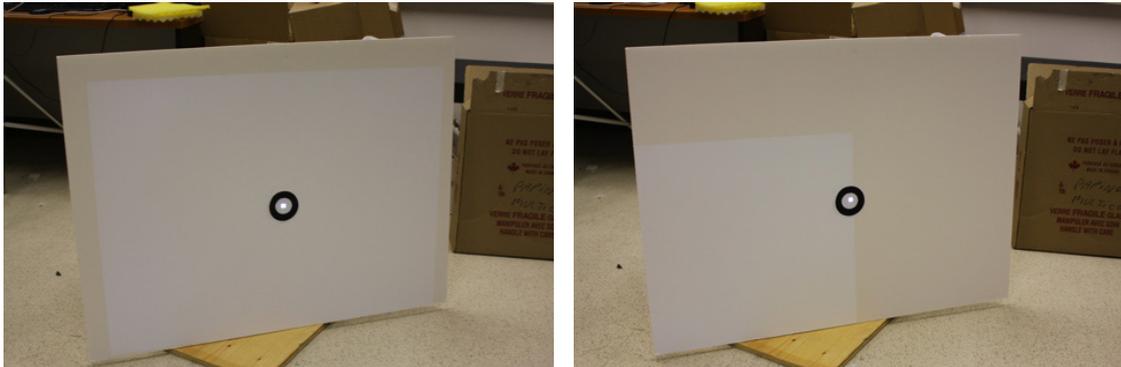


FIG. 4.21 – Projection de la grille détectée par la caméra et adaptée au point de vue du projecteur.

ce, pour différentes rotations de la tête du DLx. En pratique, puisque le projecteur ne peut pas *observer*, on observe la projection seulement dans la caméra. De plus, pour estimer l'erreur en 3D plutôt qu'en 2D, on triangule les valeurs observées en un point 3D qui sera comparé au point 3D fixe de départ. Si le DLx est calibré parfaitement, la triangulation devrait toujours reconstruire la correspondance au même point malgré le mouvement du DLx.

Les résultats présentés sont obtenus par les étapes suivantes :

- ① aligner une cible avec un pixel projecteur allumé pour créer une correspondance caméra-projecteur (figure 4.22),
- ② calculer $M_{cam} = \mathcal{K}_{cam} \mathcal{P}_{cam} \mathcal{R}_{tilt} \mathcal{R}_{pan}$ et $M_{proj} = \mathcal{K}_{proj} \mathcal{P}_{proj} \mathcal{R}_{tilt} \mathcal{R}_{pan}$ en fonction des rotations horizontale et verticale de la tête du DLx,
- ③ trianguler la correspondance et obtenir la position du point 3D fixe en fonction du DLx à l'aide de M_{cam} et M_{proj} ,
- ④ déplacer la tête du DLx en gardant le point 3D fixe dans le champ de vision de la caméra et du projecteur,
- ⑤ mettre M_{cam} et M_{proj} à jour avec les nouvelles rotations \mathcal{R}_{tilt} et \mathcal{R}_{pan} ,



(a) Correspondance caméra-projecteur pour re- (b) Reprojection du point après déplacement de
construire un point 3D servant de référence la tête robotisée

FIG. 4.22 – Reconstruction d'un point 3D à partir d'une correspondance caméra-projecteur et sa reprojection dans le projecteur après avoir subi un déplacement de la tête robotisée.

- ⑥ projeter le point 3D fixe vers la caméra et le projecteur (\mathbf{p}_{cam} et \mathbf{p}_{proj}),
- ⑦ allumer le pixel \mathbf{p}_{proj} dans le projecteur. Si la calibration est parfaite, le point sera en plein centre de la cible (voir nos résultats à la figure 4.23),
- ⑧ trouver la position de la cible dans l'image. Si la calibration est parfaite, la position de la cible aura les mêmes coordonnées que \mathbf{p}_{cam} ,
- ⑨ répéter les étapes ④ à ⑧ pour plusieurs positions de la tête.

Les tests ont été effectués sur un intervalle de 21° de rotation horizontale et de 16° de rotation verticale. La figure 4.25 montre l'étendue des angles utilisés pour le test et le résultat de la triangulation des correspondances caméra-projecteur. L'erreur 2D est illustrée à la figure 4.24. Elle est de 4,5 pixels en moyenne, mais sa distribution montre clairement que l'erreur augmente avec la magnitude du mouvement du DLx. L'erreur 3D de reconstruction qui en résulte est de $\bar{x} = 11,2 \pm 8,3$ cm.

Si la calibration était parfaite, la reprojection du point serait toujours positionnée en plein centre de la cible (figure 4.23) et la position de la cible détectée dans l'image correspondrait avec la reprojection du point 3D dans la caméra (figure 4.24).

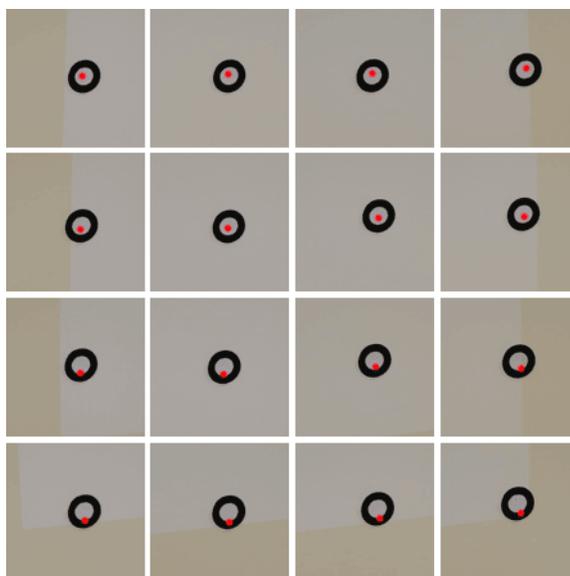


FIG. 4.23 – Reprojection du point 3D de référence dans le projecteur, après avoir appliqué des rotations horizontale et verticale à la tête robotisée.

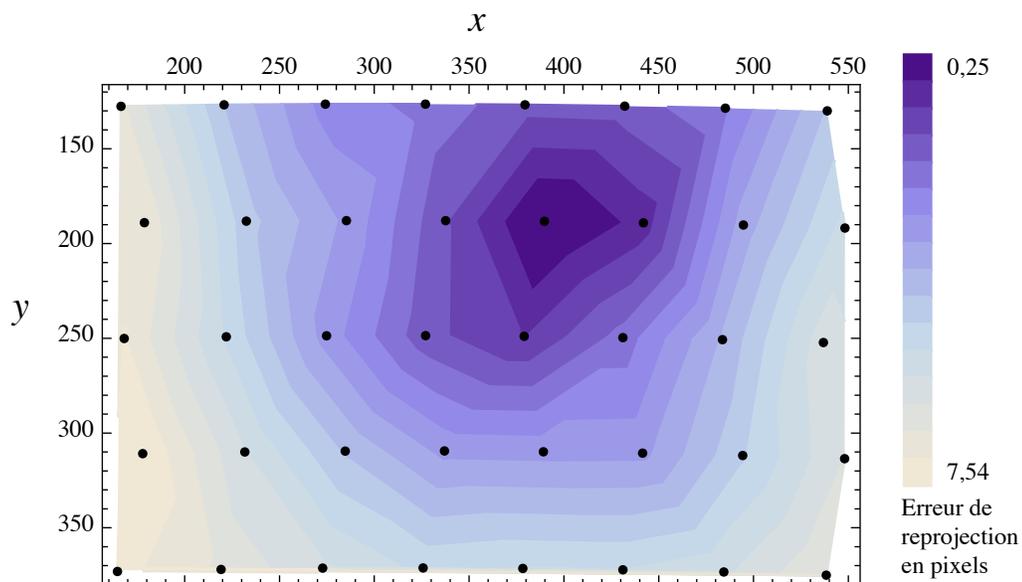


FIG. 4.24 – Détection de la cible dans l'image (noir) et distribution de l'erreur de reprojection du point 3D (gradient mauve).

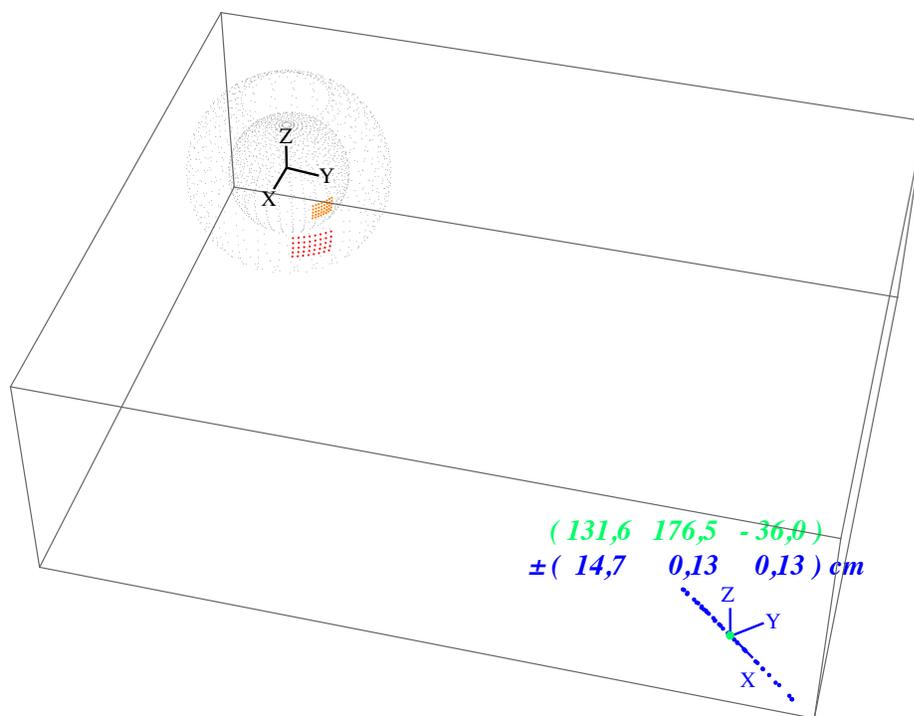


FIG. 4.25 – Visualisation des positions de la caméra (rouge) et du projecteur (orange) avec la reconstruction 3D du point de référence (vert) et les reconstructions 3D des correspondances compensées par le mouvement du DLx (bleu).

4.7 Discussion

Les tests de projection avec déplacement de la tête robotisée montrent une légère erreur de reprojection du point de référence dans la caméra et le projecteur (figures 4.23 et 4.24), qui se répercute en une grande marge d'erreur dans la triangulation des données (figure 4.25). L'erreur 3D n'est pas uniforme, elle est principalement située dans la direction de l'axe optique de la caméra. Le long de cet axe, l'erreur est de $\pm 14,67$ cm, alors qu'elle est de $\pm 0,13$ cm dans les autres directions. Cette erreur correspond à l'ajout d'un bruit aléatoire de $[-5, 5]$ pixels sur les correspondances caméra-projecteur et confirme le lien entre les erreurs 2D et 3D que nous obtenons. La magnitude de l'erreur 3D n'est pas surprenante puisque l'angle entre les rayons de projection est très petit étant donné que le plan est situé à 2 m du DLx, que la séparation caméra-projecteur est d'environ 10 cm et que les angles de vue de la caméra et du projecteur sont faibles.

L'imprécision de la projection du point de référence dans la caméra (l'erreur 2D) peut avoir été introduite par une mauvaise triangulation du point de référence ou une mauvaise calibration des axes de rotations verticale et horizontale.

Il est possible que la reconstruction 3D du point de référence soit erronée en raison d'une triangulation imprécise. Dans ce cas, les reprojections du point 3D dans le projecteur et dans la caméra seraient décalées et justifieraient les résultats obtenus. Par contre, les résultats de la section précédente montrent une bonne calibration caméra-projecteur. De plus, le résidu numérique de l'intersection des rayons de référence est faible, ce qui implique que même si les rayons ne se croisent pas parfaitement, ils sont très proches (figure 4.26). La bonne qualité des résultats étaient prévisibles à cause de la technique employée. La lumière structurée génère un grand nombre de correspondances caméra-projecteur et les points choisis couvrent un grand intervalle de profondeurs. Cela permet de contraindre énormément la solution lors de la minimisation des erreurs de reprojection (section 4.5) et d'obtenir une calibration caméra-projecteur fiable.

La reprojection du point reconstruit à chaque position de la tête montre que l'erreur est distribuée symétriquement autour du point de référence (figure 4.24). Si on met en lien les incréments de rotation utilisés et l'erreur de reconstruction 3D, on obtient le même genre de distribution symétrique d'erreur (figure 4.27). La distribution de l'erreur en fonction des rotations horizontale et verticale montre clairement la cor-

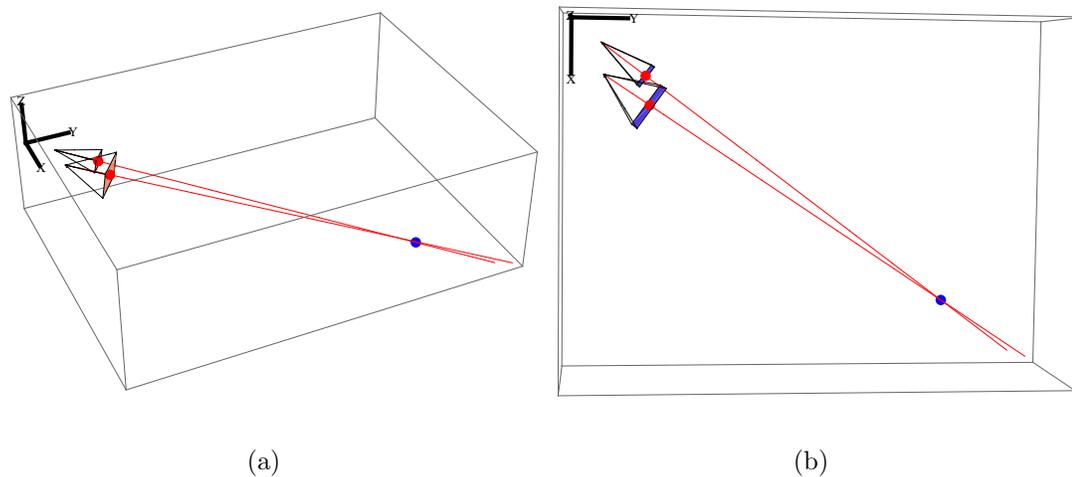


FIG. 4.26 – Visualisation de la précision de l'intersection des rayons du point de référence 3D.

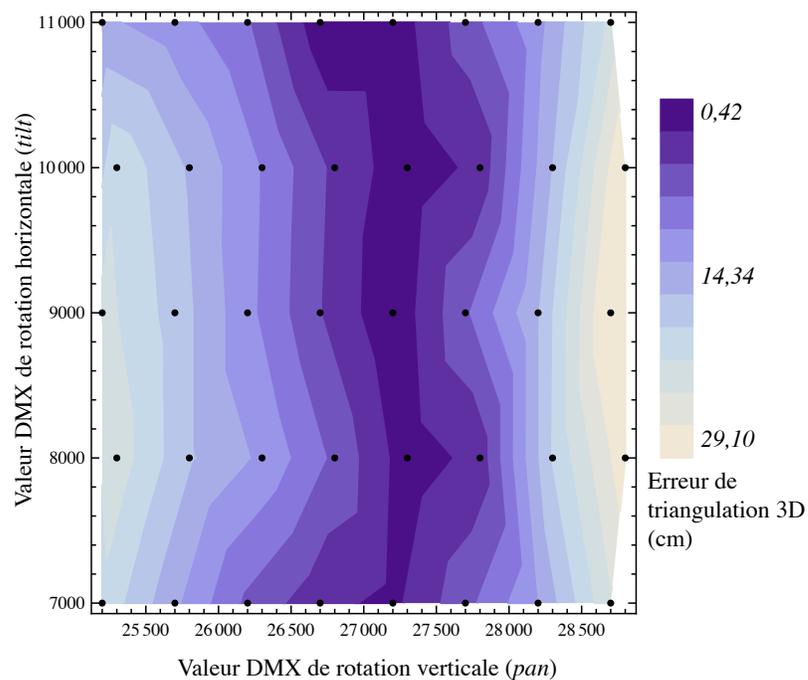


FIG. 4.27 – Visualisation de l'erreur de reconstruction 3D en fonction de la rotation de la tête robotisée (gradient mauve) et l'échantillon de reconstruction (noir).

relation entre le déplacement horizontal et l'augmentation de l'erreur. Cela confirme que la source principale d'erreur est la calibration de la pose de la caméra par rapport au centre du DLx. Étant donné que cette pose est erronée, un déplacement de la tête cause la rotation du modèle de caméra autour du mauvais axe de rotation et l'erreur s'accumule au fur et à mesure de la rotation.

Lors de la calibration de la pose de la caméra dans le monde du DLx, le résultat est ajusté en minimisant l'erreur de reprojection des plans elliptiques dans la caméra. Plutôt que d'utiliser les plans pour raffiner la pose de la caméra par rapport à l'axe de rotation horizontal, il vaudrait mieux ajouter une étape de compensation par gerbes perspectives (*bundle adjustment*) [42]. Cette méthode est fréquemment utilisée comme étape finale dans les applications de vision par ordinateur. La compensation par gerbes perspectives permet de raffiner la structure 3D des points observés et les paramètres internes et externes de caméra dans une même étape de minimisation (p.ex. l'algorithme Levenberg-Marquardt). La solution optimale minimise l'erreur de reprojection des points 3D dans la caméra (distance entre les points 3D reprojétés dans la caméra et les coordonnées observées dans l'image). Dans le cas présent, le point de référence 3D, sa reprojection dans la caméra et son observation par la caméra permettrait de retrouver avec plus de précision les poses 3D de la caméra subissant une rotation. De cette façon, la pose serait corrigée par rapport aux axes de rotation.

4.8 Application de la calibration

La modélisation du DLx a été utilisée dans un contexte réel de performance devant public. Le résultat de la collaboration entre les laboratoires Vision3D et NXI GESTATIO a été présenté dans le cadre du festival des Étés de la Danse de Paris du 21 au 23 juillet 2008. Sous la verrière du Grand Palais de Paris, trois *Tryphons* télécommandés étaient suivis par six DLx (3 DL1 et 3 DL2, voir la figure 4.28). Chaque *Tryphon* était suivi par deux DLx, ce qui permettait d'avoir une évaluation de pose du cube plus robuste et de projeter sur plus de faces visibles.

L'algorithme de suivi développé par Sébastien Roy et Nicolas Martin au laboratoire Vision3D est largement inspiré de l'algorithme CCD développé par R. Hanek [13]. La projection des coordonnées d'un cube 3D dans une image donne un contour fermé convexe. La forme du contour est directement liée à la pose du cube par rapport à la



FIG. 4.28 – Positionnement de trois DL1s sur le bord de la piste de danse.

caméra, à une rotation de 90° près. Il est possible de retrouver la pose en ajustant une courbe paramétrique définie par six paramètres de pose $(t_x, t_y, t_z, r_x, r_y, r_z)$ sur le contour du cube dans l'image. L'ajustement de la courbe sur le contour du cube dans l'image est basé sur les modèles statistiques des pixels à l'intérieur et à l'extérieur du contour. En ajustant les paramètres de la courbe pour se rapprocher du modèle statistique pour chaque image vidéo, nous obtenons directement la pose du cube volant par rapport à la caméra (exemple du suivi à la figure 4.29). Connaissant la position de la projection du cube dans la caméra, il est possible de suivre les cubes dans tout l'espace disponible puisque nous ajustons automatiquement les rotations horizontale et verticale de la tête robotisée pour que le cube reste toujours dans l'angle de vue de la caméra et du projecteur. Nous adaptons une projection sur le cube puisque nous connaissons la relation entre la caméra et le projecteur. Nous pouvons également exprimer la position absolue d'un *Tryphon* par rapport à la référence du monde puisque nous connaissons la pose de la caméra par rapport au DLx, les rotations horizontale et verticale de la tête robotisée et que les DLx sont calibrés par rapport à une référence absolue commune.

Une fois ce module de suivi développé, une interface de contrôle a été implémentée pour suivre la progression du suivi actif pour chaque DLx (figure 4.30). La piste de danse, au-dessus de laquelle les cubes volaient, était éclairée par de nombreux

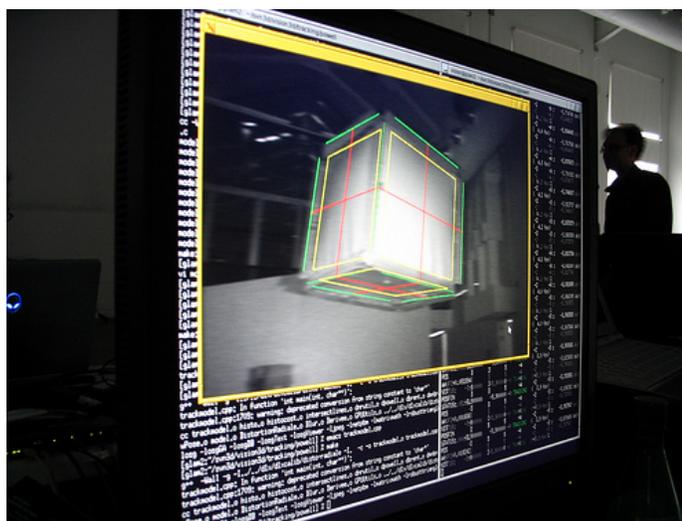


FIG. 4.29 – Détection du cube et ajustement de sa pose en utilisant les contours.

faisceaux lumineux mobiles qui inondaient parfois les cubes de lumière infrarouge et qui perturbaient le suivi des cubes. Pour contrer ce problème, l'interface nous permettait de ré-ajuster le suivi du cube lorsque celui-ci était faussé par l'éclairage. L'interface nous permettait également de contrôler le déplacement des têtes robotisées manuellement en cas de besoin. Pour s'assurer du bon suivi des cubes et de l'évaluation de la pose des cubes, le flux vidéo des six caméras était intégré dans l'interface, ainsi que la représentation en 3D du positionnement des DLx et des cubes.

Les cubes se déplaçaient dans un espace de 35 m de haut \times 40 m de largeur \times 30 m de long, ce qui demandait d'utiliser une grande étendue de la rotation des têtes robotisées. Malgré l'étendue du déplacement des têtes robotisés, les projections, dont le rendu était généré à partir du point de vue du projecteur, restaient bien alignées avec les faces visibles des *Tryphons*. Cette présentation nous a permis de juger qualitativement de la précision de la calibration des DLx.

Les vidéos mixés en temps réel par les VJs d'Elektra⁵ et le projet *Cityspeak*⁶ de Jason E. Lewis passaient par notre numériseur vidéo avant d'être projetés sur les faces d'un cube 3D. Les images projetées sur les *Tryphons* correspondaient donc à la pose et à la géométrie du cube volant et texturaient ses faces visibles (figure 4.31).

⁵ Elektra. Septembre 2009. Dernière visite : 4 octobre 2009. <http://www.elektramontreal.ca/>.

⁶ Obx Laboratories. Cityspeak. 2006. Dernière visite : 4 octobre 2009. <http://www.cspeak.net>.

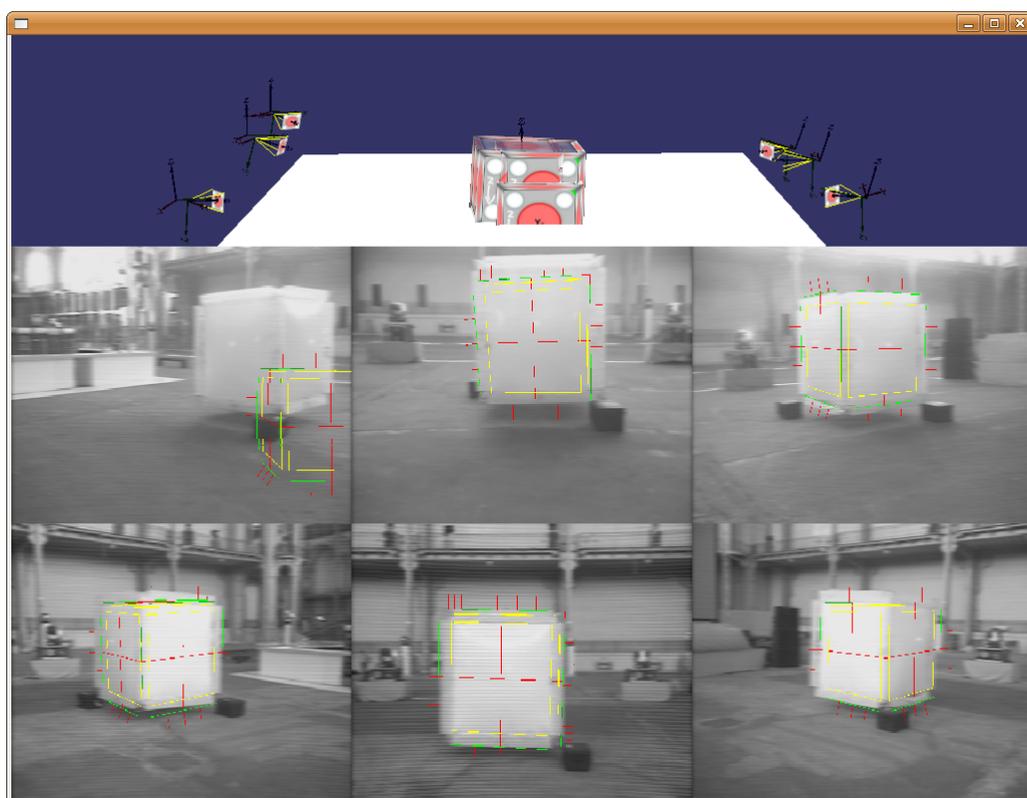


FIG. 4.30 – Capture d'écran de l'interface. On peut voir que le suivi a été mal initialisé sur les deux premières images, l'interface nous permettait de corriger ce genre de problèmes de suivi pendant la performance.

Les *Tryphons*, devenus écrans volants, ont permis aux créateurs vidéo de sortir du cadre et d'ajouter du dynamisme à la présentation de leur travail.



(a)

(b)

FIG. 4.31 – Projections des vidéos d'Elektra et de Jason E. Lewis sur les faces des *Tryphons* au Grand Palais de Paris.

Chapitre 5

(ARTICLE) VIRTUAL 3D DRAWING TOOL FOR STAGE PERFORMANCES

La Trilogie des flous est une pièce de théâtre en trois actes écrite par l'auteur et metteur en scène Daniel Danis. L'écrivain a puisé l'inspiration de cette création dans ses rêves qu'il a interprétés sous forme de poèmes avant de les adapter à la performance. Daniel Danis a décidé d'avoir recours à des projections pour mettre en scène les images évoquées dans ses poèmes et transporter les spectateurs dans un monde surréaliste. Les techniques de vision par ordinateur utilisées par le Laboratoire Vision3D ont permis d'ajouter des effets visuels encore peu utilisés dans le milieu du théâtre.

La notion d'*actant* est importante dans le théâtre de Daniel Danis. Contrairement à l'acteur qui joue un personnage selon des directives plus ou moins strictes, un actant influence le cours de la pièce en agissant au niveau narratif sur le déroulement ou au niveau physique sur son environnement. Le concept d'actant, né du théâtre contemporain et exploité par Daniel Danis, est à la base d'un théâtre expérimental interactif qui donne le pouvoir aux acteurs de manipuler l'environnement scénique visuel et sonore.

Plusieurs techniques ont été exploitées dans la pièce de théâtre. Des algorithmes de projections adaptatives ont été utilisés pour projeter des images et des vidéos sur des surfaces non-traditionnelles telles qu'un écran circulaire et un écran cylindrique (figure 5.2 page 72). Des vidéos prétraités et des vidéos des actants pris en temps réel pendant la pièce étaient projetés. La capture en temps réel permettait de réaliser de gros plans cinématographiques des acteurs, une nouvelle approche en théâtre. Les vidéos étaient également traités en temps réel pour ajouter des effets visuels au flux d'images lorsque désiré. Finalement, un outil de dessin 3D permettait à un actant de la pièce de dessiner des formes, d'écrire des mots et de contrôler des sons par le simple mouvement de son bras dans l'espace (figure 5.3 page 73). Le mouvement de l'actant était suivi et reconstruit en 3D à l'aide d'un pointeur infrarouge tenu dans la main de l'actant et de plusieurs caméras *wimotes* placées autour de la scène. Ce dernier projet

implique la calibration d'un réseau de caméra et la reconstruction 3D du tracé à partir des coordonnées images. Le projet de dessin 3D a fait l'objet d'une présentation orale à la conférence *Virtual Reality International Conference* (VRIC'09), mieux connue sous le nom de Laval Virtual [4].

La conférence regroupe plusieurs sujets appliqués dans le contexte de la réalité virtuelle. L'article a été présenté dans le cadre du symposium *Digital Art* qui s'intéresse particulièrement aux applications artistiques de la réalité virtuelle et s'articule autour de l'art visuel, la danse, le théâtre, l'architecture ou la musique, pour ne nommer que ceux-là. Présenté dans un contexte multidisciplinaire, l'article devait être accessible à la fois aux artistes et aux scientifiques présents au symposium, c'est pourquoi nous avons réduit la littérature à propos de la calibration de caméra. Puisque la calibration de caméra constitue le sujet principal de ce mémoire, il est important d'introduire les travaux connexes à cet article.

Les systèmes de surveillance et l'arrivée des environnements interactifs immersifs ont amené les chercheurs à se pencher sur la calibration multi-caméras. La calibration par plans (section 3.1) introduite par Sturm, Maybank et Zhang pour une seule caméra [36, 48, 35] est à la base de l'algorithme de calibration multi-caméras présenté par Ueshiba & Tomita [43]. Dans cet article, trois caméras observent un plan avec différentes positions et orientations. En calibrant les caméras séparément, les poses relatives des caméras devraient toujours être les mêmes peu importe les plans utilisés pour la calibration. En pratique, ce n'est pas toujours le cas à cause du bruit qu'on retrouve dans les images. Sturm [35] est le premier à incorporé cette contrainte dans un algorithme de calibration mais suppose que les paramètres internes des caméras sont déjà connus. Ueshiba & Tomita incorporent également cette contrainte mais n'exigent pas que les caméras soient pré-calibrées. Ils utilisent la méthode présentée par Zeinik-Manor & Ireni [45] pour uniformiser les facteurs d'échelles qui diffèrent pour chaque homographie et extraient la calibration et la pose de chaque caméra avec la méthode de factorisation de Sturm. Une dernière étape de minimisation non-linéaire de l'erreur de reprojection raffine le résultat de la calibration. Zhang, Li & Jia [46] réutilisent cette méthode en l'appliquant à la calibration d'un plus grand nombre de caméras qui ne voient pas tous les mêmes plans. Ils arrivent à calibrer toutes les caméras en trouvant les poses relatives entre des sous-groupes de caméras qui voient le même plan. Dans le contexte de la calibration d'un réseau de *wimotes*, la

calibration par plans requiert la construction d'un plan de calibration dont les quatre coins sont illuminés par des DELs infrarouges. Pour éviter la construction d'un objet de calibration, nous avons également exploré les possibilités de la calibration par géométrie épipolaire.



FIG. 5.1 – Manettes de contrôle de la console Wii de Nintendo (ou *wimotes*).

La calibration épipolaire n'a pas la robustesse et la précision de la calibration par plan. Pour améliorer les performances de la calibration, les algorithmes multi-caméras utilisent différentes étapes de minimisation non-linéaire qui améliorent la précision des résultats. Davis & Chen [8] utilisent les contraintes épipolaires pour retrouver une première estimation de la calibration des caméras. Les correspondances de points sont construites à l'aide d'une DEL dont le chemin couvre tout le volume de reconstruction des caméras. La géométrie épipolaire permet d'extraire la pose d'une caméra à un facteur d'échelle. Davis & Chen utilisent l'intersection des rayons des poses relatives des caméras pour établir le facteur d'échelle qui sera itérativement appliqué à toutes les caméras. Le parcours de la DEL est suivi en 3D en utilisant un filtre de Kalman étendu. La trajectoire 3D devient un objet de calibration utilisé pour itérer entre la calibration de la caméra et la reconstruction 3D de la trajectoire pour raffiner le résultat de la modélisation. Plutôt que d'utiliser un filtre de Kalman, plusieurs chercheurs ajoutent une étape de compensation par gerbes perspectives (*bundle adjustment*) [42] qui ajuste la calibration des caméras [20, 37, 5]. La calibration multi-caméras par géométrie épipolaire ne requiert qu'un minimum d'étapes à l'utilisateur

et n'impose pas la construction d'objet de calibration complexe. Dans le contexte d'une performance théâtrale, la facilité d'utilisation du processus de calibration est très importante, c'est pourquoi nous avons opté pour cette technique.

La suite du chapitre présente l'article suivant dans sa version originale : Lucie BÉLANGER et Sébastien ROY, « Virtual 3D drawing for stage performances ». Dans *Proceedings of Virtual Reality International Conference*, Laval, France, Avril 2009.

Abstract

This paper presents a new system for tridimensional virtual drawing relying on an infrared pen observed by a number of surrounding cameras. Nintendo wiimotes are used as cameras to provide reliable real-time tracking information. Since 3D triangulation requires the pen to be visible from at least 2 cameras, which highly constrains the field of view, we support many wiimotes for accurate reconstruction over a large space, while providing maximum movement flexibility to a performer. The paper focuses on the calibration of multi-wiimotes setups and accurate 3D reconstruction of trajectories. Tested with up to 6 wiimotes and used in real performance context, the system provides both accurate 3D tracking and great performance flexibility.

Keywords : 3D tracking, wiimote, camera calibration, epipolar geometry

5.1 Introduction

Interest in wiimote applications grew rapidly after the launch of the product by Nintendo. Many developers put effort into reverse engineering the communication protocol to take advantage of the sensor technology used in the wiimote. Using infrared reflection/emission and the wiimote's camera, the range of projects suddenly expanded. It democratized the development of new user interfaces based on infrared tracking, allowing people to create their own living room interactive displays or whiteboards, based on the work by Johnny Chung Lee¹. Most of the applications developed incorporate single-view 3D geometry concepts. Using two wiimotes, an infrared plane and an infrared pointer, people developed a real-time 3D reconstruction application based on a planar calibration [17].

Due to our application in stage performances and the limited coverage of the previous technique, we developed a calibration application that integrates any number of wiimotes, thus allowing a much larger reconstruction space. This extension required to develop our own calibration procedure that was adapted to the wiimotes but flexible enough to account for many wiimote configurations. The result was a stage performance tool that allowed performers to create and manipulate virtual content in real-time.

¹ Johnny Chung Lee. Johnny Chung Lee - Projects - wii. Last checked : October 4th, 2009. <http://www.cs.cmu.edu/~johnny/projects/wii/>.

The creative context of the development of this application influenced many technical decisions in the final implementation. The following sections detail the 3D geometry concepts that were necessary to build the application and how we adapted them to our application. Finally, a thorough analysis of our reconstruction results allowed us to test the accuracy of the system.

5.1.1 *Technology in theater performances*

The starting point of this project is a collaboration between a computer vision laboratory and a play writer. The director and writer Daniel Danis works to create a new kind of production where stage performers can control their environment by manipulating sounds, images, videos and the physical supports of these medium. Text, staging and choreography suddenly become canvas for the actors instead of restraining them.

This project falls within a long list of interactive performance work developed in the last decade. Technology infiltrated the performance art scene as more projects started to successfully create bridges between the body and the environment (physical and/or virtual). Dancers and actors could experiment with their bodies and extend their abilities to controlling lighting, soundscapes and projections. These augmented performance spaces forced choreographers to think about new ways of defining creative languages [6]. David Rokeby played a major role in the development of optical tracking in digital art with his *Very Nervous System* [31] and helped defining the baseline of interactive techniques [32]. Although the use of wearable sensors is the most common approach used, many projects use optical tracking in dance [19, 10], music [26] and theater [34].

The writing tool was developed during creation workshops that gathered the writer, the composer, the actors and the developers. It was used during public performances of *La Trilogie des Flous* at the *Usine C* Montreal, Canada, and at the *Studio national du Fresnoy* in Tourcoing, France. During the performance, the tool was used to write words that were strongly evoking the poems that inspired the trilogy (see figure 5.3 and figure 5.16 for an illustrated example). The paths drawn on the stage were used to control the sampling of the narrator's voice, and the sound granulation of chosen voice extracts.

Developing technological tools with artists for artists has its own set of constraints



(a) Configuration of the stage prior to the performance.

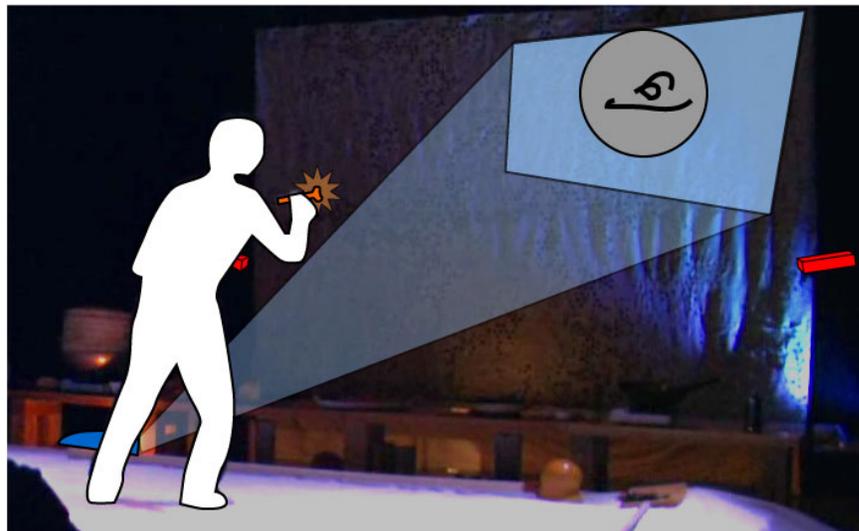


(b) During the performance, videos were projected on a cylindrical screen (left) and on a small circular screen (far right) using adaptative projection techniques [39].

FIG. 5.2 – Stage configuration of *La Trilogie des Flous* as presented at *Usine C* in Montreal.



(a) Scene detail during *Reneiges*, the third scene in the trilogy



(b) For clarity's sake, the actor is outlined in white, the infrared pen in orange and the wiimotes in red. A projector (in blue) was hidden under a decor table and, using adaptative projection techniques [39], was projecting the drawings in real-time on a circular screen (in gray).

FIG. 5.3 – Daniel Danis drawing on a circular screen with the tool during a performance of *La Trilogie des Flous* at *Usine C* in Montreal.

and difficulties. Feedback is an important issue in the process since artists are used to see the results of their work on a short-time scale. Programmers would often favor scripting languages and graphical programming environment over a compiled language in order to gain speed and prototyping capabilities. The tools developed have to be flexible. Artists do not want the technology to tell them what to do, they want to use technology as a medium. Finally, not all actors, dancers or visual artists are used to computers and gadgets so you want these tools to be very intuitive. All these constraints are quite challenging to consider in a experimental development context, but the final result gain a lot a robustness from them.

5.1.2 *Working with wiimotes*

The context influences the design of the tools, but also the equipment used to achieve it. Some tracking systems were developed specifically for drawing in 3D space or track actors on stage. Most of the drawing tools developed use magnetic tracking ². Although it allows a precise reconstruction, the reconstruction space is very limited and drawing is usually in the range of a forearm movement. Many artists have used these systems for 3D graphics drawing [44, 22] or for stereoscopic animation using the SANDDE interface [25, 18] developed by IMAX ³. These systems, aside from being expensive, were not appropriate in our case because their reconstruction space is limited and they are designed as software drawing tools rather than real-time performance tools. Immersive optical tracking systems are another option ⁴, but their price makes them impossible to use in small experimental productions.

We wanted our tracking system to be affordable, easy to setup, flexible and easily available to the public. For these reasons, using wiimotes as infrared cameras was a natural choice for this prototype. Creators often have to work with really small budget and wiimotes are the most affordable camera you can find on the market. Integrating technology into a stage performance is often cumbersome so wireless technology is often a plus. Since fast prototyping is a big issue, having the image processing done directly in the wiimotes saves lots of time and can avoid many potential errors.

² Ascension Technology Corporation. Flock of birds tracking system. Last checked : October 4th, 2009. <http://www.ascension-tech.com/realtime/RTflockofBIRDS.php>.

³ IMAX. Corporate profile. Last checked : October 4th, 2009. <http://www.imax.com/corporate/>.

⁴ Motion Capture Systems from Vicon. Last checked : October 4th, 2009. <http://www.vicon.com>.

Moreover, the high frame rate of the wiimote makes it a perfect solution for tracking. Finally, many open source libraries and softwares, like `cwiid`⁵ and `libwiimote`⁶, are available to easily communicate with the wiimotes.



FIG. 5.4 – Custom-made wide-angle infrared pointer using 16 infrared LEDs and two AA batteries.

Work was put into the infrared device. During the first set of presentations of the play, the actor was using a candle but the low intensity of the flame was forcing a smaller range of reconstruction. For the second set of performances, we developed a infrared pen using high-power (100mA) infrared LEDs having a 60° diffusion angle and emitting a 940 nm wavelength. Since LEDs only emit in one direction, the pen was not visible in more than two wiimotes at the time which makes calibration hard in a 360° environment. To solve the problem, we built a multi-LED prototype pen using 16 LEDs, powered with 2 AA batteries (see figure 5.4). The result gave a infrared pen with a 180° emitting angle on a 360° axis (see figure 5.5) with a range of over 5 meters from the wiimote. We ran optical tests using an infrared sensitive camera in order to measure the performance of the prototyped pen. We could not use the wiimotes for the tests since we cannot have access to the images. The tests confirmed that the

⁵ CWiid. Last checked : October 4th, 2009. <http://abstrakraft.org/cwiid/>.

⁶ Simple Wiimote Library for Linux. Last checked : October 4th, 2009. <http://libwiimote.sourceforge.net/>.

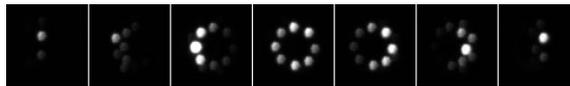


FIG. 5.5 – Infrared visibility of the pen position at angles -90°, -60°, -25°, 0°, 25°, 60° and 90° from the camera optical axis.

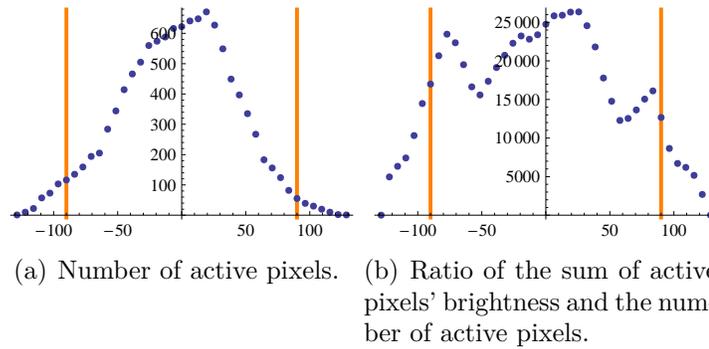


FIG. 5.6 – Graphs representing the brightness of the IR pen for angles between -130° et 130° (90° angles are represented with orange lines).

two concentric LED rings allow a wider coverage. Figure 5.6(a) shows the number of active pixels (a pixel was considered active when its brightness was greater than 8% of its maximum brightness). We notice that we can detect the pen over slightly more than 180° . Figure 5.6(b) shows the ratio between the total brightness of the active pixels over the number of active pixels. The form of the plot is representative of the alignment of the LEDs at the tip of the pen. Using multiple high-power LEDs placed in two concentric rings allows us to gain a better range of reconstruction, higher precision and gives more flexibility to the actor.

5.2 Multiple wiimote calibration

The calibration system requires a minimum of two wiimotes. More wiimotes can be used to cover the space required by the application, as long as their coverage overlaps. The maximum number of wiimotes is the number of Bluetooth connections a computer can handle simultaneously. Our test setup was composed of four wiimotes placed around a half circle (later tests were performed with six wiimotes in different configurations, see figure 15 for examples).

The use of wiimote cameras limits the possibilities of the calibration method to be used. Planar calibration requires building a calibration device [17], which complicates the setup. Since we only have access to the 2D coordinates of a maximum of four infrared sources, 3D points calibration cannot be easily applicable neither. On the other hand, using wiimotes makes it easier to use epipolar geometry constraints since we have direct correspondences with very low probability of outliers, so this is our

avored approach.

A prerequisite step of calibration is to select a camera order which maximize the overlap, or number of common points between successive cameras. In what follows, we will assume that cameras 1, 2, 3, ... have been ordered. In most cases, this corresponds to the placement order around the performance stage.

The internal calibration process, described in section 5.2.1, determines the camera viewing angle and is manually computed once per wiimote. An initial calibration based on epipolar geometry is performed independently for each pair of cameras in the system. It is described in section 5.2.2. These initial poses are fed into a minimization algorithm, described in section 5.2.3, that will enforce the 3D reconstruction continuity across cameras. The 3D reconstructions resulting from this calibration will be expressed into a global arbitrary scale. We can perform a last calibration step in order to express the 3D data into a particular unit system. Our calibration approach is very similar to the one presented by P. Baker and Y. Aloimonos [2]. The main difference between our approaches is that they force a full 360° coverage of the cameras to get better calibration accuracy while we favor flexibility over accuracy.

In order to align all cameras in the same scale and coordinate system, it is required to have 3D points visible from at least three cameras. These common points will help enforce proper cameras alignment during the minimization step. It is also important to sample the whole performance space, since this will ensure a better calibration.

5.2.1 *Internal calibration*

Setting up a fully calibrated multi-camera system requires to know minimal information about the optics of the camera. In this case, very few details are provided by the manufacturer. The developer community figured the resolution to be 1024×768 and the viewing angle to be 45°.

A calibration application requires to know the camera center and the focal length of a camera. Although each camera is slightly different, we estimated the camera center to be at (512, 384). The viewing angle was computed experimentally for each remote by measuring the ratio $f = nb$ of pixels * distance/length. The results we got were very similar for each wiimote, $fx = 1340.43$ and $fy = 1318.15$, which gives a horizontal viewing angle of about 42°. These experimental results proved to be precise enough for our system.

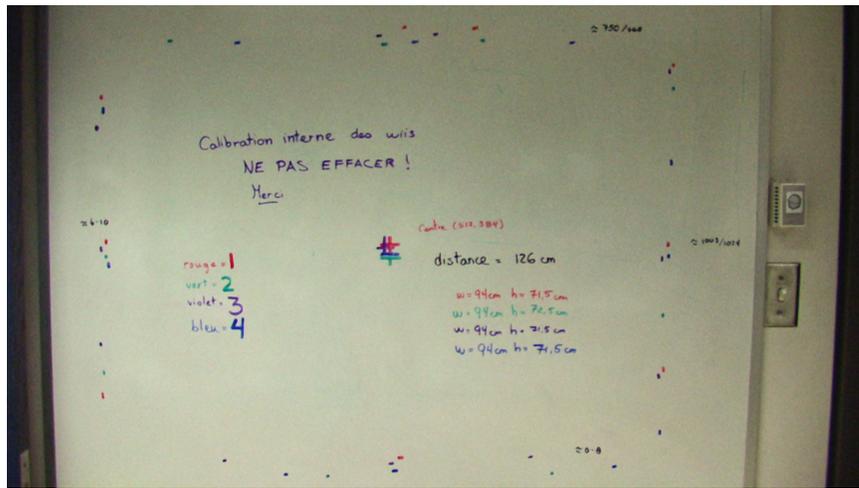


FIG. 5.7 – Each wiimote’s viewing angle was calibrated independently by marking the edges of the field of view on a whiteboard at a known distance.

5.2.2 Pose from epipolar geometry

Epipolar geometry represents the relation between two views. Since we are working with more than two views, the external calibration has to be done in a few steps. For each consecutive pair of cameras for which we have point correspondences, we can compute the fundamental matrix that represents the relation between the two cameras [23, 14]. Since we have an estimation of the internal parameters of the wiimotes, we can derive the essential matrix from the fundamental matrix and extract the pose of one of the cameras relatively to its neighbor.

The fundamental matrix is found using the Ransac algorithm on the set of points. Although the probability of outliers is very low due to the use of the wiimote processing, we used Ransac in order to make sure that no unsynchronized coordinate would influence the calibration. We do not know the order in which the data is received from the wiimotes, which means that some points could be mistakenly associated to the wrong frame.

Once every consecutive pairs of camera has been calibrated, each camera has a local pose. We want to express the camera poses according to the same reference point. In our case we simply used the first camera as the reference, although the reference could be a virtual camera. By multiplying the relative poses of the cameras we can express all cameras in an absolute system according to the first camera, or to

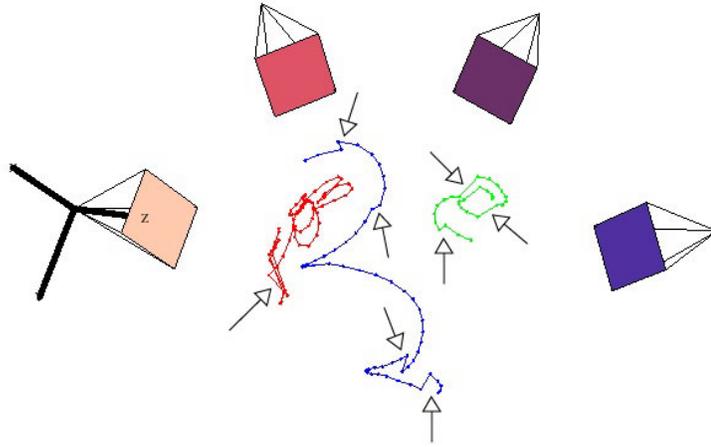


FIG. 5.8 – Even though the camera positions look good, the 3D reconstruction shows major misalignments (shown with arrows).

any other reference desired.

Since each pair of cameras is calibrated independently, we cannot ensure the continuity of the 3D reconstruction between each pair of cameras. The result of calibration using epipolar geometry is up to a scale factor. Two pairs of cameras might not be calibrated on the same scale, which means that a point visible from three cameras might not be reconstructed at the same position when triangulated with the first two cameras or with the last two. In order to solve the problem, we need 3D points that are visible from at least three cameras. These multiple-view points allow to adjust the scale of each pose and align the cameras more precisely through a minimization process.

5.2.3 Pose minimization

The goal of the calibration algorithm is to be flexible to different wiimote configurations. The only assumption made is that each neighboring wiimotes has overlapping view fields. We do not make any assumptions about the number of wiimotes or the visibility of points, although scaling adjustment will require having 3D points visible from as at least three cameras. The pose minimization has to be done in multiple passes in order to account for any wiimote configurations.

Each camera c has six parameters to minimize, the translation \mathbf{T}^c expressed at

an unknown scale factor and the rotation \mathbf{R}^c , combined as a 3×4 matrix \mathbf{M}^c . Each minimization step proceed over a subset s, \dots, t of cameras. Each evaluation of the cost function $f^{s;t}$ in eq. 5.1 re-triangulates the 3D coordinates $P^{s;t}$. Each updated 3D coordinate $P_i^{s;t}$ is reprojected in every camera $c = s, \dots, t$ as \hat{x}_i^c which is compared to the original image point x_i^c .

The cost function used in the minimization is the sum of the reprojection errors of all the triangulated points into the visible cameras. The reprojection error is the square distance between the reprojected point and the original coordinate.

The following pseudo-code summarizes the steps of the algorithm. Given N wii-motes, the minization proceeds over subsets of wii-motes. Each subset includes cameras s to t . During the minimization only part of those cameras can change (cameras a to t only). The minimisations are carried in sequence following the camera order ($s = 1, 2, 3, \dots$) and increasing subset size ($m = 1, 2, 3, \dots$).

Algorithm 1 Pose optimization procedure

```

for  $m = 2, 3, \dots, N$  do
  for  $s = 1, 2, \dots, N - m + 1$  do
     $t = s + m - 1$ 
     $a = \lfloor \frac{s+t}{2} \rfloor + 1$ 
    Minimize  $f^{s;t}(\mathbf{M}^s, \dots, \mathbf{M}^t)$  over  $\mathbf{M}^a, \dots, \mathbf{M}^t$  using Levenberg-Marquardt
  end for
end for

```

where

$$f^{s;t}(\mathbf{M}^s, \dots, \mathbf{M}^t) = \sum_{c=s}^t \sum_{i \in V^{s;t}} D(x_i^c, \hat{x}_i^c) \quad (5.1)$$

and

$$\begin{aligned}
 V^{s;t} &= \{i | x_i^c \text{ is visible in every camera } c = s, \dots, t\} \\
 \hat{x}_i^c &= \mathbf{M}^c P_i^{s;t} \\
 P_i^{s;t} &= \text{triangulate}(\{\mathbf{M}^c\}_{c=s, \dots, t}, \{x_i^c\}_{c=s, \dots, t})
 \end{aligned}$$

$D(u, v)$ is the Euclidean distance between 2D projective points u and v . Note that this algorithm must be adapted to handle a camera cycle by wrapping appropriately the camera indices.

We do not make assumptions about the visibility of points but we want the minimization to take advantage of it. Multiple-view points are gradually integrated into the minimization to gain reconstruction precision. The minimization process described above is illustrated in figure 5.9.

5.2.4 Unit system transformation

The 3D reconstruction obtained from the calibration process is not expressed into a tangible unit. We can easily bring the system back into a metric system by figuring out the scaling factor between the current unknown unit and the metric system. To do so, we simply measured the distance between the first camera and all the others in metrics. By comparing these results with the camera positions obtained from the minimization, we find the scaling factor that allows to transform the reconstruction results into metric distances.

In our setup, the distance between the first and the last wiimote was measured to be 244 cm. The position found from the calibration was $(-0.93382, -0.413258, 1.5846)$ which gives a distance of 1.88514 from the first wiimote placed in $(0, 0, 0)$. From these values, we get a scaling factor of 223.856. Expressing our reconstruction in the metric system allows a better interpretation of the practical results.

5.3 Results

Two different datasets were collected from the testing setup. The first one, containing a total of 386, points was used for calibration. The second consisted of simple 3D forms performed in the space in order to analyze the quality of the calibration. In both cases, some points were visible by only two cameras, others by three cameras, and some were visible from all four cameras. We ran our minimization algorithm and reconstruction tool on different subsets of these datasets to analyze the quality of our system. We compared reprojection errors, relative reconstruction errors and qualitative visual results.

The mean reprojection error computed on the calibration set was 3.5 pixels, on a 1024×768 image. On the test dataset, the error was higher, 4.5 pixels, but close enough to validate the calibration. In figure 5.11, we notice that most points match perfectly with the original coordinates but some are off, mainly in the reprojection in

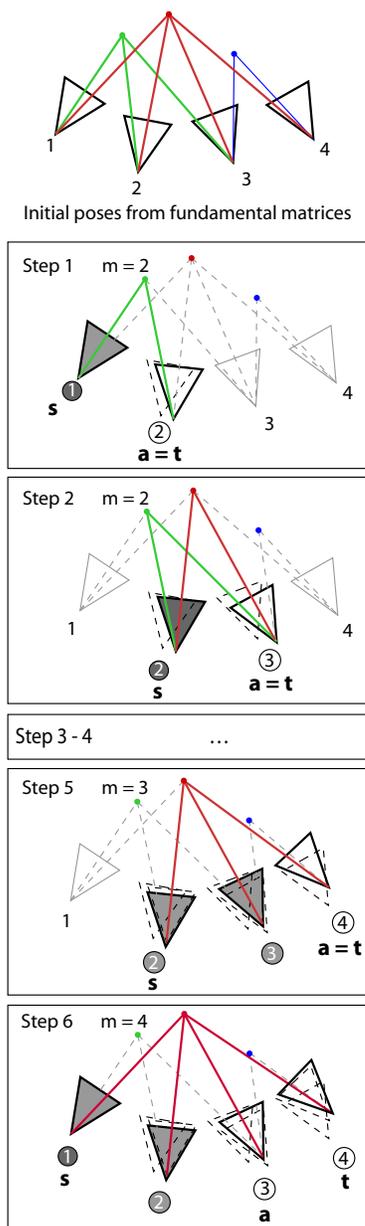


FIG. 5.9 – Illustration of the minimization process. Dark grey = fixed camera, white = minimized camera, light gray = not considered at this step.

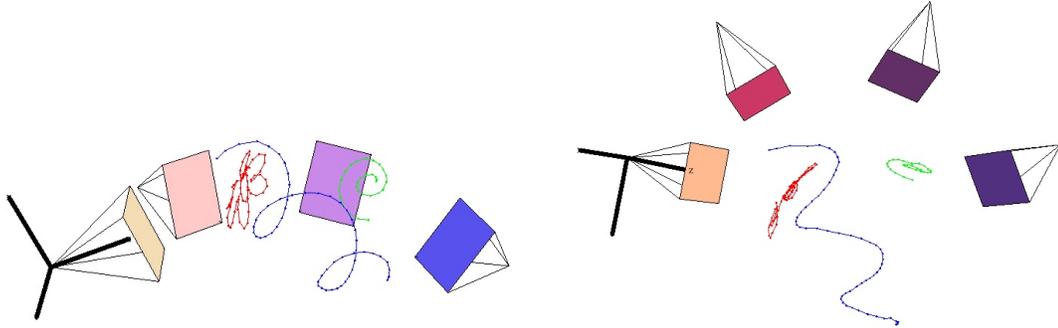


FIG. 5.10 – Poses obtained after complete calibration seen from two viewpoints. Compared to figure 5.8 the 3D reconstruction is continuous.

the first camera. A possible explanation would be the infrared pen and the wiimote's processing. The image processing done on the wiimote returns the centroid of the infrared blob found in the image. The large tip of our infrared pen emits at an angle of 180° on a 360° axis. Not all LEDs are seen by a camera when the pen moves. From figure 5.5, we can see how the LEDs' restrictive angles can affect the coordinates of the centroid returned by a wiimote and cause the reprojection error we see in figure 5.11(a). Finally, since we cannot ensure the order of the incoming coordinates, it is possible that we mistakenly match a camera coordinate to coordinates from other cameras in a different frame, although within a very small time range since the wiimotes have a frame rate of 100Hz.

From the reprojection error, we can also conclude that the use of the Ransac algorithm is not useful in our application. The very few outliers encountered can influence the fundamental matrix solution slightly, but they do not differ enough from the dataset to influence the minimization. Even if the fundamentals are computed with the small outliers, the minimization will correct the poses (see figure 5.12).

We compute a relative reconstruction error on our datasets because we were not interested in absolute measurement precision. The concern with our application was the continuity of the 3D triangulation between the cameras. We want any subsets of cameras to triangulate a point at the same 3D position to avoid reconstruction gaps between subsets of cameras. In order to quantify this error, we use points that are visible from as much cameras as possible and we triangulate the 2D coordinates with

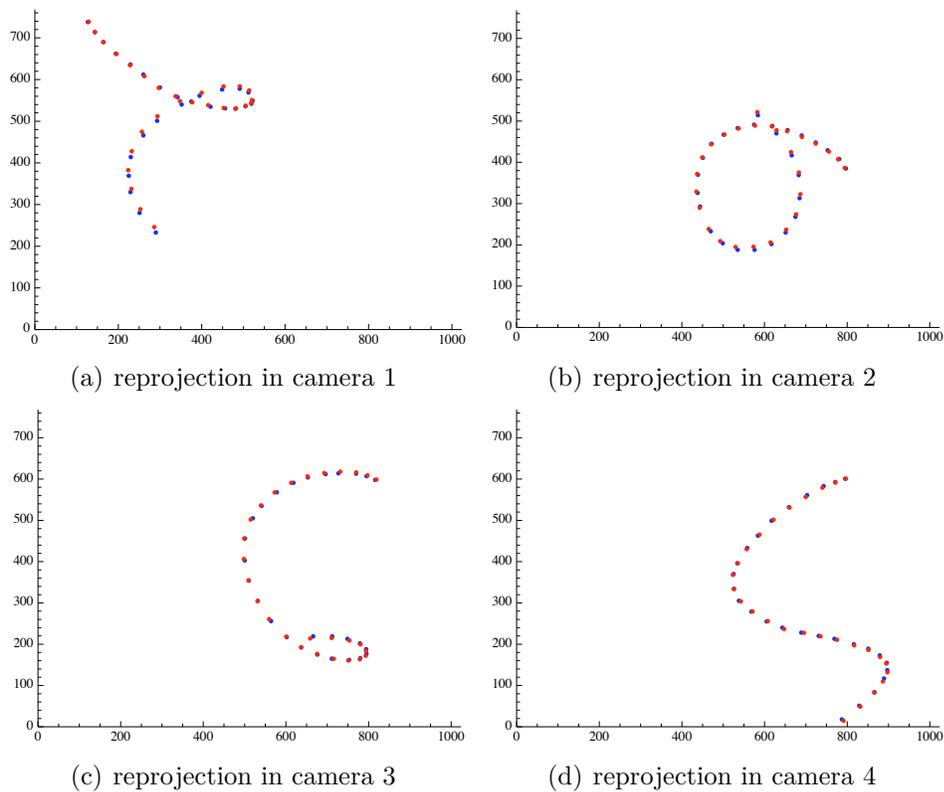


FIG. 5.11 – Reprojection (in red) of reconstructed 3D points visible from all cameras. Original world coordinates are shown in blue.

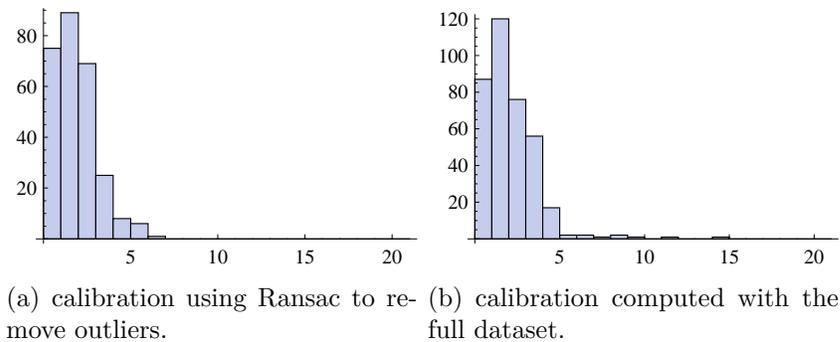


FIG. 5.12 – Histogram of the reprojection errors (in pixels) illustrating the effect of Ransac.

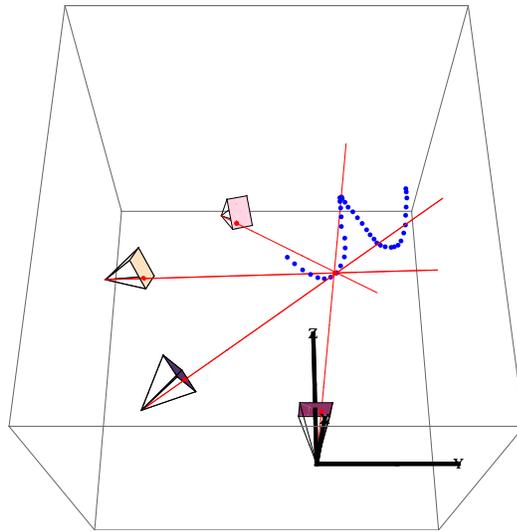


FIG. 5.13 – Example of 3D triangulation showing intersecting rays for a point visible from four wiimotes.

different camera subsets. Once we have multiple reconstruction solution for the same point, we compute the mean distance of all combinations of solutions. The result of our calibration gave a relative reconstruction error of 2.1 cm in a 33m^3 reconstruction volume (cylinder of 4.6 m diameter and 2 m height). By looking at the IR pen morphology in figure 5.4, it is hard to expect better results since 2 cm is roughly the distance between two facing LEDs from the exterior ring of the pen tip. Prototyping a smaller IR pen would most probably enhance the reconstruction accuracy. From figure 5.14, we notice a very important enhancement of the reconstruction when we use 3D visible from 3 cameras as compared to 2-view points. On the other hand, we also notice that the integration of more visible points in the minimization does not have a high impact on the accuracy of reconstruction in our case. The 3D reconstruction is a ray intersection computation (see figure 5.13). Since rays rarely intersect perfectly, the solution is often an average of the rays which helps accounting for small errors in the system calibration and favor a smooth continuity of reconstruction in our case.

Finally, the visualization of the 3D reconstruction of the test dataset is a good indication of the degree of precision of the calibration. Figure 5.15 shows that multiple-view points helps the reconstruction precision. In figure 5.15(a) we can notice major misalignments in the spiral reconstruction that are corrected throughout the minimi-

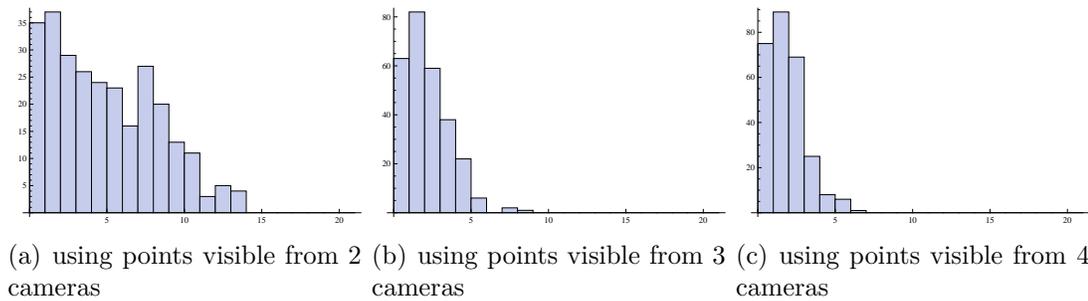


FIG. 5.14 – Relative 3D reconstruction error (in cm) for calibration results with 3D points of different visibility.

zation until the final result in figure 5.15(b).

5.4 System implementation

5.4.1 Data acquisition

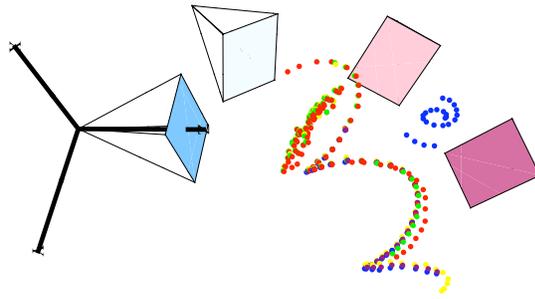
We use the *CWiid* library ⁷ in order to communicate with the wiimotes. The infrared values received through the library are sent directly into a Pure Data patch ⁸ we programmed to collect and/or process the information. Calibration data is saved on disk to be used by our C minimization program. Once the system is calibrated, any coordinates received from at least two wiimotes will be triangulated automatically for a real-time reconstruction.

5.4.2 Calibration program

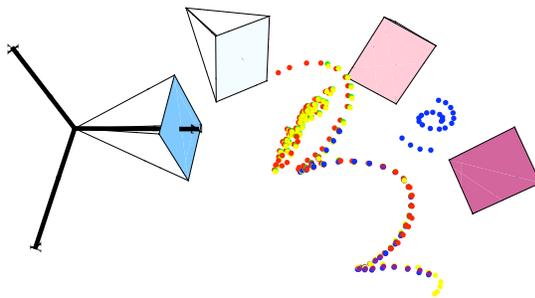
The external calibration described earlier is programmed in C. The minimization is computed using the Levenberg-Marquadt algorithm provided by the *cminpack* C package. All other algorithms (Ransac, fundamental computation, triangulation, etc.) are from our own computer vision library written in C. The complete calibration, including minimization, of our test setup using 386 points and four wiimotes runs below a minute.

⁷ *CWiid*. Last checked : October 4th, 2009. <http://abstrakraft.org/cwiid/>.

⁸ Miller Puckette and al. Pure Data - PD Community Site. October 4th, 2009. <http://www.puredata.org>.



(a) calibration only minimizing pairs of cameras



(b) calibration taking advantage of multi-view points

FIG. 5.15 – 3D reconstruction on all points with different subsets of cameras.

5.4.3 *Real-time triangulation*

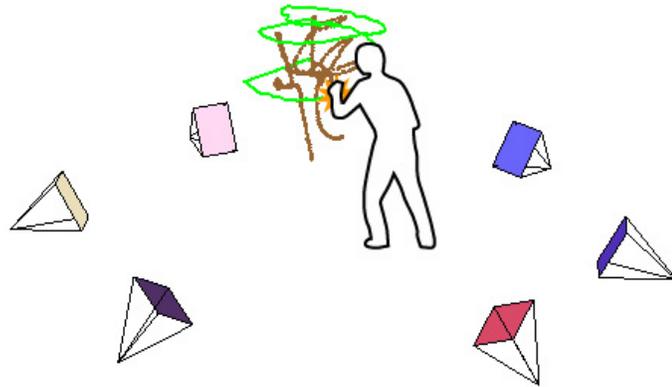
In order to allow a flexible use of the reconstruction data, we wrote a Pure Data [triangulation] object in C. Since the object is written in C, the triangulation is easily computed in real-time. In practice, we figured that a reconstruction frame rate of 20Hz was fast enough to follow the actors movement very accurately and light enough to be multi-casted through an already busy local network. Having the triangulation as a Pure Data objects allows a very flexible use of the tool. Data could be sent to another application on the computer that would process the result on screen. It could also be sent over the network for the electronic composer to use the data, or to another computer to manipulate video.

Since our application was mostly dedicated to writing and drawing in space, we put a lot of work into the feel of the tool. We wanted the actors to be comfortable with the tool so that they could easily experiment with it. In that sense, we decided to filter all incoming wiimote coordinates through a low-pass filter by converting the 3D data into a Pure Data signal. This was necessary in order to minimize shaky movements of the user. With a reconstruction error of two centimeters, any movement of the wrist is visible, even the undesired one. To facilitate to use of the pen during the performance, the scale of the reconstruction visualization projected on the screen was adaptative. The actors could produce drawings the size of a hand, or words that would cover the whole stage. In either case, the projection was adapting in order to provide the actors with high flexibility of movements and to provide the spectators with a good visual experience.

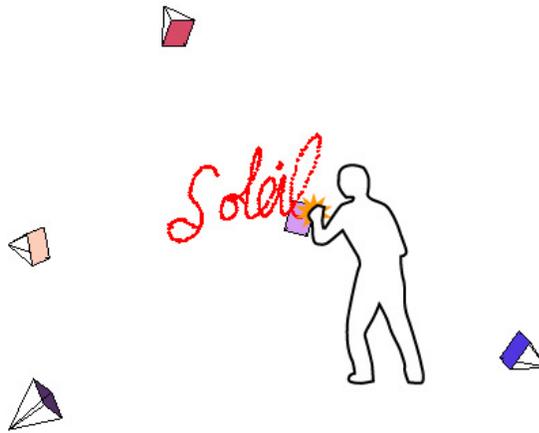
5.5 *Conclusion*

This paper presented a working large-scale 3D tracking system based on multiple Nintendo wiimotes. A new calibration approach was proposed to tackle the difficult problem of accurately reconstructing accross camera boundaries.

Our experimental reconstruction tests achieved 2.1 cm mean 3D relative error between consecutive cameras, over a 4.6 m diameter area. The tracking system was used successfully in a real performance setup, thereby proving its practical usefulness. The use of epipolar geometry and multiple steps of pose minimization makes the calibration algorithm able to handle different wiimote configurations, both in numbers and



(a) A 360° setup of wiimotes allows the actor to move freely in the space without having to worry about blocking the field of view.



(b) A configuration that allows higher reconstruction space by putting the wiimotes at different heights

FIG. 5.16 – Examples of different wiimote configurations.

spatial disposition. The implementation as a PureData triangulation object makes it easy to apply the 3D tracking system to new applications.

In the future, the 3D drawing tool will be integrated in our 360° immersive screen technology, the Cyclorama. A new version of the Cyclorama, currently in development, will feature 360° stereoscopic projections that will allow new kinds of 3D immersive interaction.

Many implementation decisions were made to enhance the accessibility of our tool to the community. In order to allow the development of new types of tracking applications with the wiimote, we planned to release an open source version of the software in a near future. New types of interactions and user interfaces will surely derive from this wiimote 3D tracking project.

Chapitre 6

CONCLUSION

La calibration de caméra permet l'utilisation de caméras et de projecteurs pour augmenter et interagir avec l'environnement. J'ai présenté deux types de calibration de caméra et leurs applications en arts visuels et en art performatif. La calibration des systèmes caméra-projecteur de type DLx rend possible le suivi actif et la projection sur des surfaces dynamiques. L'utilisation et la calibration d'un réseau de *wimotes* permet de reconstruire une trajectoire 3D dans un grand volume à un prix beaucoup plus abordable que les systèmes multi-caméras conventionnels.

La calibration des DLx a été présentée en plusieurs étapes. La contrainte de col-linéarité de la géométrie projective a permis de paramétrer la distorsion radiale de la caméra. Les paramètres internes ont été calibrés à l'aide de plans. Nous avons proposé une nouvelle méthode pour calibrer la pose d'une caméra sur tête robotisée en générant des plans elliptiques. Ces informations ont permis de raffiner les paramètres internes estimés du projecteur et de calibrer la pose du projecteur en utilisant les contraintes de la géométrie épipolaire et des correspondances caméra-projecteur provenant de la lumière structurée. La calibration des DLx a permis d'augmenter le projet SAILS de Nicolas Reeves et de transformer les *Tryphons* en canevas volants pour des artistes en arts visuels et pour des performances.

La méthode proposée pour calibrer un réseau de *wimotes* est basée sur les contraintes de la géométrie épipolaire. Cette technique permet à un utilisateur de calibrer facilement et précisément plusieurs *wimotes* et de les utiliser pour faire de la reconstruction 3D. L'accessibilité des caméras et d'un logiciel de calibration démocratise les applications de reconstruction de trajectoire 3D. Le but du développement était l'intégration de l'outil dans un contexte théâtral. L'*idéateur* du crayon 3D, Daniel Danis, a incorporé des *wimotes* à la scène pour permettre à l'actant de dessiner et de contrôler les paramètres d'extraits sonores.

6.1 Applications futures

Les travaux présentés sont à la base de plusieurs développements futurs. La calibration des DLx pourrait être plus précise en utilisant plusieurs plans elliptiques concentriques pour couvrir un plus grand angle de rotation horizontale. En ajoutant une étape de compensation par gerbes perspectives avec un point 3D connu et sa re-projection dans la caméra et le projecteur, on pourrait raffiner le résultat de la pose. En ayant une meilleure reconstruction 3D, nous pourrions utiliser ce système pour connaître précisément la pose d'un *Tryphon* dans l'espace et contrôler automatiquement sa trajectoire de vol en communiquant directement avec l'ordinateur de bord. Étant donné l'application dans le domaine des arts, il serait avantageux d'automatiser davantage le système de calibration. Plusieurs étapes de calibration pourraient être combinées pour faciliter la prise des données de calibration, minimiser la répercussion des erreurs dans les étapes subséquentes et accélérer le processus de calibration du modèle complet.

Nous avons intégré l'outil de dessin 3D dans l'écran cylindrique immersif du laboratoire Vision3D (annexe B), tel que mentionné dans le chapitre 5. L'écran et la configuration des *wimotes* permettent de dessiner en 3D avec vision stéréoscopique dans un espace de 360° autour de l'utilisateur. Plusieurs nouvelles applications du système de dessin 3D sont en cours, avec le metteur en scène Daniel Danis, dans un nouveau projet de théâtre intitulé *Yukie*. Un algorithme de reconnaissance de gestes sera ajouté pour contrôler plus de paramètres sonores et permettre des interactions plus riches avec les écrans de projection.

Les présentations publiques réalisées au Grand Palais (Paris), au Studio national des arts contemporains le Fresnoy (Tourcoing) et à l'Usine C (Montréal) ont démontré le potentiel des collaborations entre les arts visuels, l'art performatif et la vision par ordinateur. La réception positive des projets contribue au développement et à la continuation des collaborations en cours.

RÉFÉRENCES

- [1] ASHDOWN, Mark et Yoichi SATO. « Steerable projector calibration ». Dans *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*. IEEE Computer Society, Washington (DC), États-Unis, 2005, p. 98.
- [2] BAKER, Patrick et Yiannis ALOIMONOS. « Complete calibration of a multi-camera network ». Dans *Proceedings of the IEEE Workshop on Omnidirectional Vision (OMNIVIS)*. IEEE Computer Society, Washington (DC), États-Unis, 2000, p. 134–145.
- [3] BASU, Anup et Kavita RAVI. « Active camera calibration using pan, tilt and roll ». Dans *IEEE International Conference on Robotics and Automation*, tome 3. IEEE Computer Society, Washington (DC), États-Unis, 1995, p. 2961–2967.
- [4] BÉLANGER, Lucie et Sébastien ROY. « Virtual 3D drawing tool for stage performances ». Dans *Proceedings of Virtual Reality International Conference*. Laval, France, avril 2009.
- [5] BÉRIAULT, Sylvain, Pierre PAYEUR et Gilles COMEAU. « Flexible multi-camera network calibration for human gesture monitoring ». Dans *International Workshop on Robotic and Sensors Environments (ROSE)*, 2007, p. 1–6.
- [6] BIRINGER, Johannes. « Dance and interactivity ». *Dance Research Journal*, tome 35/36(2/1), 2003, p. 88–112.
- [7] BUTZ, Andreas et Christian SCHMITZ. « Annotating real world objects using a steerable projector-camera unit ». Dans *CVPR '05 : Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops*, tome 3. IEEE Computer Society, Washington (DC), États-Unis, 2005, p. 109.
- [8] CHEN, Xing, James DAVIS et Philipp SLUSALLEK. « Wide area camera calibration using virtual calibration objects ». Dans *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'00)*, tome 2. IEEE Computer Society, Washington (DC), États-Unis, 2000, p. 520–527.

- [9] COLLINS, Robert et Yanghai TSIN. « Calibration of an outdoor active camera system ». Dans *IEEE Computer Vision and Pattern Recognition (CVPR '99)*, tome 1. IEEE Computer Society, Washington (DC), États-Unis, juin 1999, p. 528–534.
- [10] CONIGLIO, Mark et Dawn STOPPIELLO. *Troika Ranch*. Dernière visite le 4 octobre 2009. <http://www.troikaranch.org/>.
- [11] DAVIS, James et Xing CHEN. « Calibrating pan-tilt cameras in wide-area surveillance networks ». Dans *IEEE International Conference on Computer Vision*. IEEE Computer Society, Washington (DC), États-Unis, 2003, p. 144–149.
- [12] EHNES, Jochen, Koichi HIROTA et Michitaka HIROSE. « Projected augmentation - augmented reality using rotatable video projectors ». Dans *IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, Washington (DC), États-Unis, 2004, p. 26–35.
- [13] HANEK, Robert et Michael BEETZ. « The contracting curve density algorithm : Fitting parametric curve models to images using local self-adapting separation criteria ». *International Journal of Computer Vision (IJCV)*, tome 59, 2004, p. 233–258.
- [14] HARTLEY, Richard I. « In defense of the eight-point algorithm ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, tome 19(6), 1997, p. 580–593.
- [15] HARTLEY, Richard I. et Andrew ZISSERMAN. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York (NY), États-Unis, 2e édition, 2004.
- [16] HARTMAN, Jacob et Tony LEVAS. « Interacting with steerable projected displays ». Dans *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*. IEEE Computer Society, Washington (DC), États-Unis, 2002, p. 402.
- [17] HAY, Simon, Joseph NEWMAN et Robert HARLE. « Optical tracking using commodity hardware ». Dans *ISMAR '08 : Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*. IEEE Computer Society, Washington (DC), États-Unis, 2008, p. 159–160.

- [18] HETHERINGTON, Janet. *The National Film Board of Canada : Auteur Animation*. *Animation World Magazine*. 28 novembre 2007. <http://www.awn.com/articles/national-film-board-canada-iauteuri-animation>.
- [19] HU, Song, Jesper MORTENSEN et Bernard F. BUXTON. « A real-time tracking system developed for an interactive stage performance ». Dans *World Academy of Science, Engineering and Technology*, tome 5, avril 2005, p. 102–105.
- [20] IHRKE, Ivo, Lukas AHRENBERG et Marcus MAGNOR. « External camera calibration for synchronized multi-video systems ». Dans *WSCG 2004 : the 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision 2004 ; short communication papers proceedings, Journal of WSCG*, tome 1. UNION Agency, Plzen, République Tchèque, février 2004, p. 537–544.
- [21] JUNEJO, Imran N. et Hassan FOROOSH. « Practical pure pan and pure tilt camera calibration ». Dans *19th International Conference on Pattern Recognition (ICPR)*, décembre 2008, p. 1–4.
- [22] KEEFE, Daniel F., Daniel A. FELIZ, Tomer MOSCOVICH, David H. LAIDLAW et Joseph LAVIOLA. « Cavepainting : A fully immersive 3D artistic medium and interactive experience ». Dans *Proceedings of the 2001 Symposium on Interactive 3D graphics*. ACM Press, New York (NY), États-Unis, 2001, p. 85–93.
- [23] LONGUET-HIGGINS, Hugh Christopher. « A computer algorithm for reconstructing a scene from two projections ». *Nature*, tome 293(5828), 1981, p. 133–135.
- [24] LOWE, David G. « Distinctive image features from scale-invariant keypoints ». *International Journal of Computer Vision (IJCV)*, tome 60(2), 2004, p. 91–110.
- [25] NATIONAL FILM BOARD OF CANADA. *Techniques : Animation et ordinateur*. 2006. Dernière visite le 4 octobre 2009. <http://www3.onf.ca/animation/objanim/fr/techniques/animation-ordinateur.php>.
- [26] NG, Kia C. « Music via motion : transdomain mapping of motion and sound for interactive performances ». *Special Issue, The Proceedings of the IEEE*, tome 92(4), 2004, p. 645–655.
- [27] PAVIS, Patrice et Christine SHANTZ. *Dictionary of the theatre : terms, concepts, and analysis*. University of Toronto Press, janvier 1999.

- [28] PINGALI, Gopal, Claudio PINHANEZ, Anthony LEVAS, Rick KJELDSSEN, Mark PODLASECK, Han CHEN et Noi SUKAVIRIYA. « Steerable interfaces for pervasive computing spaces ». Dans *PERCOM '03 : Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*. IEEE Computer Society, Washington (DC), États-Unis, 2003, p. 315–322.
- [29] PINHANEZ, Claudio. « Augmenting reality with projected interactive displays ». Dans *Proceedings of International Symposium on Virtual and Augmented Architecture (VAA '01)*. Dublin, Irlande, 2001.
- [30] PINHANEZ, Claudio S. « The everywhere displays projector : A device to create ubiquitous graphical interfaces ». Dans *UbiComp '01 : Proceedings of the 3rd international conference on Ubiquitous Computing*. Springer-Verlag, Londres, Royaume-Uni, 2001, p. 315–331.
- [31] ROKEBY, David. *Very Nervous System*. 1986-1990. Dernière visite le 4 octobre 2009. <http://homepage.mac.com/davidrokeby/vns.html>.
- [32] ROKEBY, David. *Digital illusion : entertaining the future with high technology*, chapitre The construction of experience : interface as content. ACM Press/Addison-Wesley Publishing Co., New York (NY), États-Unis, 1998, p. 27 – 47.
- [33] SINHA, Sudipta N. et Marc POLLEFEYS. « Towards calibrating a pan-tilt-zoom camera network ». Dans *OMNIVIS 2004, 5th Workshop on Omnidirectional Vision, Camera Networks and Nonclassical cameras*, 2004.
- [34] SPARACINO, Flavia, Glorianna DAVENPORT et Alex PENTLAND. « Media in performance : interactive spaces for dance, theater, circus, and museum exhibits ». *IBM Systems Journal*, tome 39(3-4), juillet 2000, p. 479–510.
- [35] STURM, Peter F. « Algorithms for plane-based pose estimation ». Dans *IEEE Conference on Computer Vision and Pattern Recognition*, tome 1. IEEE Computer Society, Washington (DC), États-Unis, 2000, p. 706–711.
- [36] STURM, Peter F. et Stephen J. MAYBANK. « On plane-based camera calibration : A general algorithm, singularities, applications ». Dans *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, tome 1. IEEE Computer Society, Washington (DC), États-Unis, 1999, p. 432–437.

- [37] SVOBODA, Tomáš, Daniel MARTINEC et Tomáš PAJDLA. « A convenient multi-camera self-calibration for virtual environments ». *Presence : Teleoperators and Virtual Environments*, tome 14(4), août 2005, p. 407–422.
- [38] TARDIF, Jean-Philippe et Sébastien ROY. « A MRF formulation for coded structured light ». Dans *Proceedings of the Fifth International Conference on 3-D Digital Imaging and Modeling*. IEEE Computer Society, Washington (DC), États-Unis, 2005, p. 22–29.
- [39] TARDIF, Jean-Philippe, Sébastien ROY et Martin TRUDEAU. « Multi-projectors for arbitrary surfaces without explicit calibration nor reconstruction ». Dans *Fourth International Conference on 3D Digital Imaging and Modeling (3DIM)*, 2003, p. 217–224.
- [40] TARDIF, Jean-Philippe, Peter F. STURM et Sébastien ROY. « Self-calibration of a general radially symmetric distortion model ». Dans *Lecture Notes in Computer Science*, tome 3954. Springer Berlin/Heidelberg, 2006, p. 186–199.
- [41] TRAJKOVIĆ, Miroslav. « Interactive calibration of a ptz camera for surveillance applications ». Dans *Asian Conference Computer Vision*. Melbourne, Australie, janvier 2002.
- [42] TRIGGS, Bill, Philip MCLAUCHLAN, Richard HARTLEY et Andrew FITZGIBBON. « Bundle adjustment - a modern synthesis ». Dans *Proceedings of the International Workshop on Vision Algorithms (ICCV)*. Springer-Verlag, Londres, Royaume-Uni, 2000, p. 298–372.
- [43] UESHIBA, Toshio et Fumiaki TOMITA. « Plane-based calibration algorithm for multi-camera systems via factorization of homography matrices ». Dans *Proceedings of Ninth IEEE International Conference on Computer Vision*, tome 2. IEEE Computer Society, Washington (DC), États-Unis, 2003, p. 966–973.
- [44] WANG, Charlie C. L. « Drawing on air : Input techniques for controlled 3D line illustration ». *IEEE Transactions on Visualization and Computer Graphics*, tome 13(5), 2007, p. 1067–1081.
- [45] ZEINIK-MANOR, Lihi et Michal IRANI. « Multiview constraints on homographies ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, tome 24(2), février 2002, p. 214–223.

- [46] ZHANG, Liuxin, Bin LI et Yunde JIA. « A plane-based calibration for multi-camera systems ». Dans *IEEE International Conference on Computer-Aided Design and Computer Graphics*. IEEE Computer Society, Washington (DC), États-Unis, 2007, p. 365–370.
- [47] ZHANG, Zhengyou. *A New Multistage Approach to Motion and Structure Estimation : From Essential Parameters to Euclidean Motion Via Fundamental Matrix*. Rapport technique, INRIA, 1996.
- [48] ZHANG, Zhengyou. « Flexible camera calibration by viewing a plane from unknown orientations ». Dans *International Conference on Computer Vision (ICCV)*, 1999, p. 666–673.
- [49] ZHANG, Zhengyou. « A flexible new technique for camera calibration ». *IEEE Transactions on Pattern Analysis and Machine Intelligence*, tome 22(11), novembre 2000, p. 1330–1334.

Annexe A

PROJECTIONS ADAPTATIVES SUR LES TRYPHONS

Les *Tryphons* du projet SAILS (*Self-Aligning-Lighter-than-Air-Structure*), de l'artiste Nicolas Reeves et du laboratoire NXI GESTATIO de l'UQÀM, se sont avérés être des robots particulièrement intéressants comme surface de projection. Le laboratoire Vision3D de l'Université de Montréal, dirigé par le professeur Sébastien Roy, s'est penché sur la problématique de la projection sur une surface cubique en mouvement. Le résultat de la collaboration a donné naissance à un écran volant intelligent capable d'interagir avec son environnement.

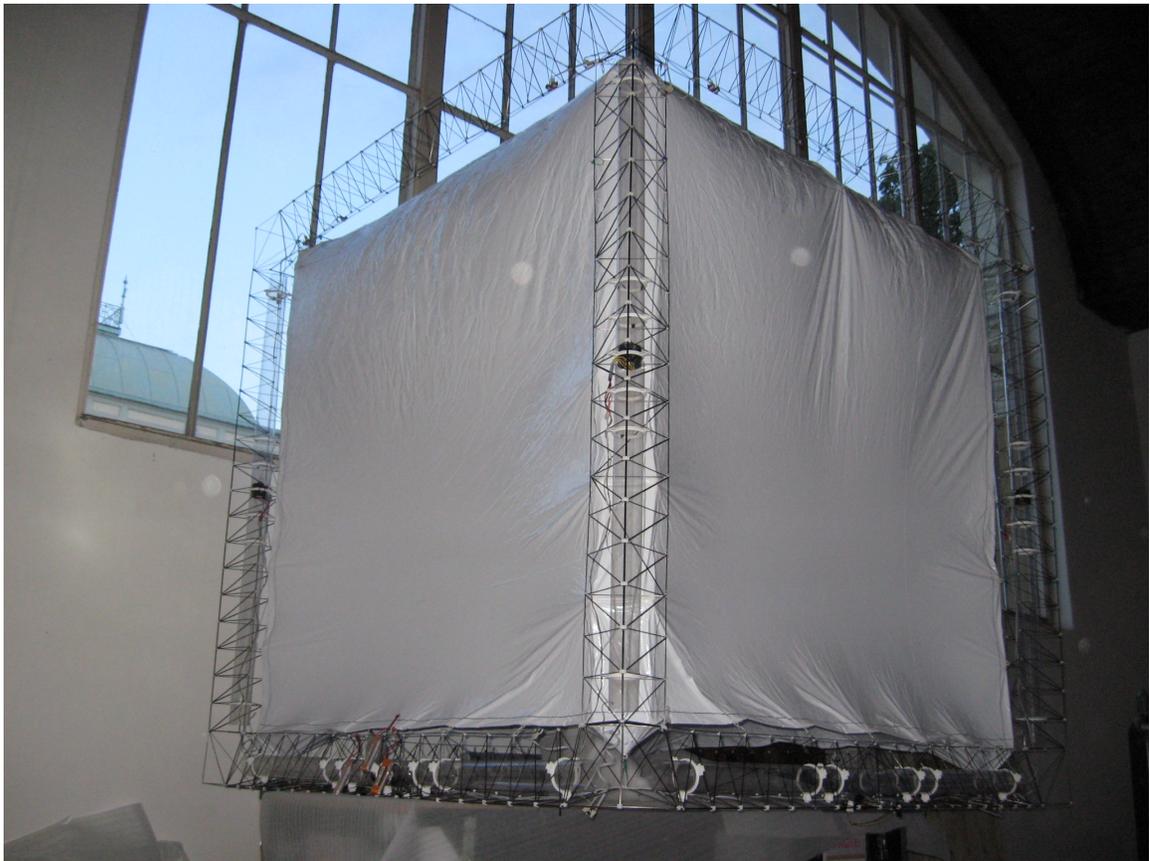


FIG. A.1 – Morphologie d'un *Tryphon*.

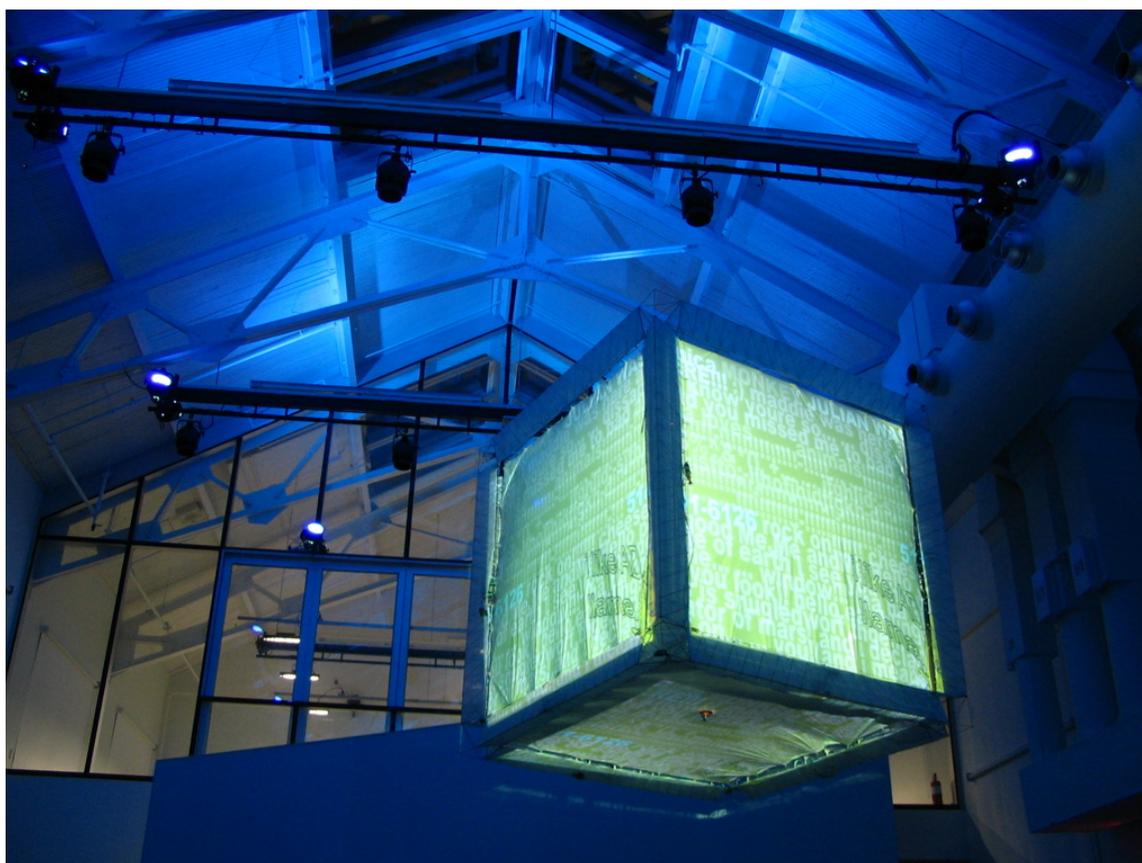


FIG. A.2 – Test de projection sur un *Tryphon* au Coeur des Sciences de l'UQÀM. La vidéo projetée est le projet *Cityspeak*, de Jason E. Lewis, qui reçoit et affiche en temps réel les messages textes envoyés par des utilisateurs anonymes.

Annexe B

DESSIN 3D IMMERSIF

L'outil de dessin 3D et de calibration multi-caméras utilisant des *wimotes* a été intégré dans l'écran immersif stéréoscopique du laboratoire Vision3D. Le positionnement des manettes sur le pourtour de l'écran permet de couvrir tout l'espace intérieur du Cyclorama et de visualiser les dessins immersifs tridimensionnels créés avec l'outil.

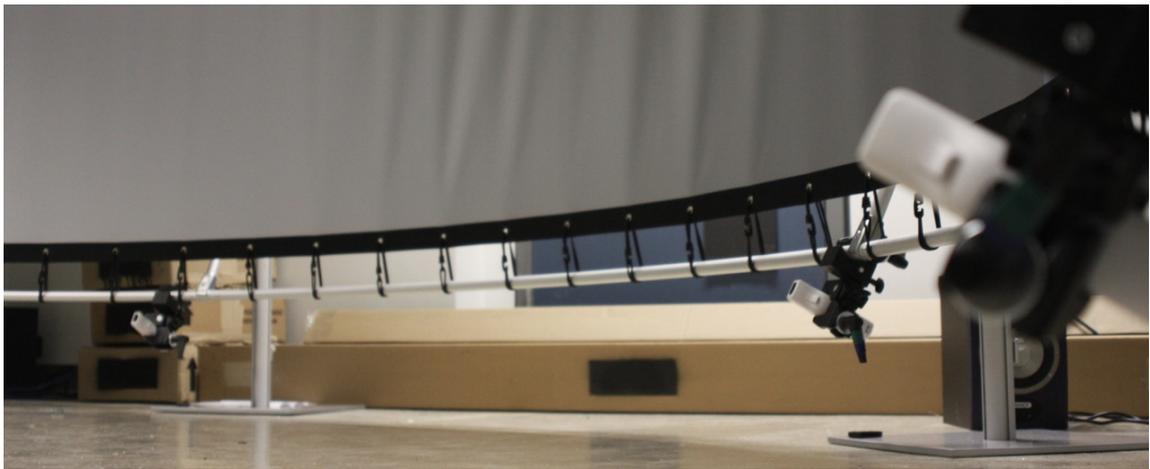


FIG. B.1 – Positionnement des *wimotes* autour de l'écran immersif.

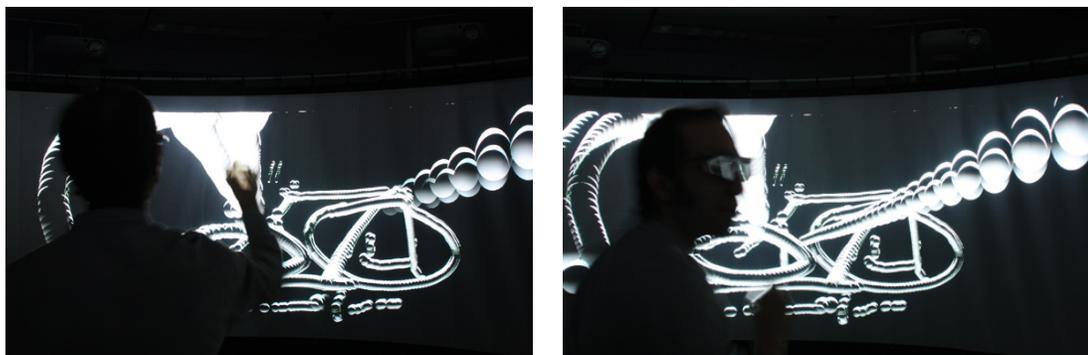


FIG. B.2 – Dessin stéréoscopique et visualisation à l'aide de lunettes polarisées.

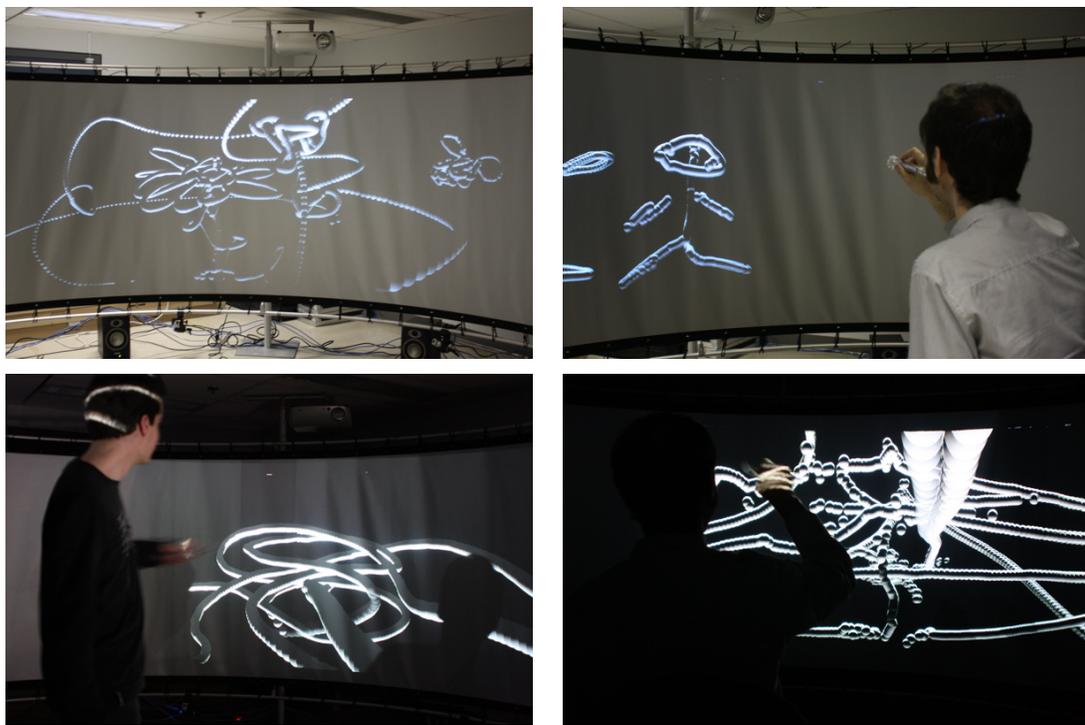


FIG. B.3 – Tests de dessins 3D immersifs en temps réel.