

---

## Etude de faisabilité d'un système de métrologie de miroirs déployables pour un CubeSat par déflectométrie.

**Auteur :** Jehasse, Lindsay

**Promoteur(s) :** Habraken, Serge

**Faculté :** Faculté des Sciences

**Diplôme :** Master en sciences physiques, à finalité approfondie

**Année académique :** 2019-2020

**URI/URL :** <http://hdl.handle.net/2268.2/9292>

---

### *Avertissement à l'attention des usagers :*

*Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.*

*Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.*

---



Faculté des Sciences  
Département de physique

---

# Etude de faisabilité d'un système de métrologie de miroirs déployables pour un CubeSat par déflectométrie

---

JEHASSE Lindsay

Promoteur : HABRAKEN Serge

Travail de fin d'études présenté  
en vue de l'obtention du grade de  
**Maître en Sciences Physiques**

Année académique 2019-2020

# Remerciements

Tout d'abord, je souhaiterais remercier mon promoteur le professeur Serge Habraken pour m'avoir offert l'opportunité de travailler sur un sujet qui me plaisait vraiment. De plus, ses conseils ainsi que ceux de Jean-François Vandenrijt furent d'une grande aide pour mener à bien ce mémoire. Je remercie plus particulièrement leurs encouragements lors des périodes de doutes de ma part.

Je remercie le Centre Spatial de Liège de m'avoir accueilli à bras ouverts et surtout m'avoir fourni un bureau et un laboratoire.

Je voudrais remercier tout spécialement ma maman et Grégory Baltus pour leur soutien indéfectible durant cette année. Je pouvais compter sur eux à n'importe quel moment. Leurs relectures, leurs opinions et leurs réconforts furent une véritable source d'énergie.

Je remercie également Vincent Boudart et Kevin Jehasse pour m'avoir permis d'utiliser leur logiciel Matlab afin d'acquérir quelques données. Je remercie plus particulièrement Kathleen Jacquerie pour m'avoir appris à élaborer un code sur Matlab.

Plus généralement, je tenais à remercier tous les gens qui m'entourent. Ils ont tous vécu directement ou indirectement aux rythmes de mes études et leurs intérêts à cet égard me touchent énormément.

Je conclus ces remerciements en spécifiant que mes années d'études à l'Université de Liège resteront à jamais de très bons souvenirs. J'ai rencontré des personnes formidables.

*En réalité, chaque lecteur est, quand il lit, le propre lecteur de soi-même. L'ouvrage d'un écrivain n'est qu'une espèce d'instrument optique qu'il offre au lecteur afin de lui permettre de discerner ce que sans le livre il n'eût peut-être pas vu en soi-même.*

Marcel Proust

# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Notions préliminaires et état de l'art</b>	<b>3</b>
1.1 Les CubeSats	3
1.2 Structures déployables	5
1.3 Métrologie optique	9
1.3.1 Méthodes Interférométriques	10
1.3.2 Méthode non interférométrique : la déflectométrie	10
<b>2 Mise en oeuvre au laboratoire</b>	<b>27</b>
2.1 Erreurs systématiques	27
2.1.1 Erreur sur la phase	27
2.1.2 Erreurs liées à l'écran	29
2.1.3 Erreurs liées à la caméra	31
2.2 Matériel	33
2.3 Configuration	36
2.4 Calibration	37
2.4.1 Géométrie	38
2.4.2 Caméra	38
2.4.3 Ecran	39
<b>3 Ray tracing</b>	<b>43</b>
3.1 Choix de la technique	43
3.2 Algorithme de ray tracing	43
3.3 Algorithmes de traitement des données	50
3.3.1 Enroulage de la phase	50
3.3.2 Création d'un masque	51
3.3.3 Déroulage de la phase	51
3.3.4 Polynômes de Zernike	55
3.4 Alignement parfait	56
3.5 Robustesse de l'algorithme de déroulage	61
3.5.1 Pixels morts	62
3.5.2 Bruit aléatoire local	63
3.6 Désalignement individuel	64
3.6.1 Elaboration d'un modèle pour une translation en y	64
3.6.2 Translation inconnue	67

<b>4</b>	<b>Perspectives et conclusion</b>	<b>68</b>
4.1	Perspectives . . . . .	68
4.1.1	Algorithmes . . . . .	68
4.1.2	Traitement des données . . . . .	68
4.1.3	Mise en oeuvre . . . . .	69
4.2	Conclusions . . . . .	69

# Introduction

De nos jours, le secteur spatial atteint son apogée. L'intérêt croissant dans ce domaine, encore fort inconnu, exige de relever beaucoup de défis. Pour accéder à cet univers, les progrès technologiques ne cessent de croître, ainsi que la taille des satellites. Les coûts liés aux projets spatiaux sont dès lors, de plus en plus conséquents. C'est dans cette problématique qu'apparaît l'idée d'élaborer des CubeSats. En effet, ces nanosatellites de par leur taille demandent un coût de fabrication beaucoup plus faible, les rendant accessibles aux universités et aux firmes privées. Cependant, leur taille impose une miniaturisation des éléments constituant la charge utile. Des structures déployables font donc leur apparition pour déplier les éléments autour du satellite. Cette technologie ouvre la voie vers des satellites de plus en plus grands, sans que le volume disponible dans la coiffe du lanceur ne devienne un problème. Lorsqu'il s'agit de déploiement de miroirs, ceux-ci n'ont pas le droit d'être positionnés n'importe comment. En effet, le déploiement est une phase critique où l'alignement des miroirs doit être extrêmement précis. Une technique de métrologie pour CubeSat doit absolument être mise en place pour mesurer ces écarts de positions afin de les rectifier à la précision voulue.

Pour mesurer les positions de miroirs déployables, l'interférométrie et la déflectométrie sont deux candidats. De par sa nature insensible aux vibrations et sa facilité de mise en oeuvre, la déflectométrie pourrait être sérieusement envisagée. De plus, une étude de la déflectométrie au niveau CubeSat est une première.

Le but de ce mémoire est d'étudier la possibilité de corriger l'emplacement de miroirs sur un CubeSat au moyen de la déflectométrie. Il ne s'agit donc pas de donner une réponse catégorique à la question, mais plutôt d'élaborer tout un cheminement vers la construction d'un système de mesures et d'envisager si le dispositif pourrait ou non être un candidat pertinent au travers de diverses analyses.

Ce mémoire se découpe en quatre chapitres, chacun ayant son propre objectif. **Le chapitre 1** traite tout d'abord de la notion de CubeSat ainsi que des structures de déploiement. Ce petit détour hors de la physique est primordial afin de s'appropriier le domaine dans lequel la technique de déflectométrie doit être mise en oeuvre. En effet, certaines restrictions du contexte pourraient dès le début proscrire la déflectométrie. Suite à cette prise de connaissance, nous nous remettons à la déflectométrie pour des miroirs sur un CubeSat afin de mesurer le déploiement de ceux-ci. La suite du chapitre fera un état de l'art sur les différentes implémentations de la déflectométrie. Ainsi, nous pourrions choisir celle qui est la plus adaptée à notre problématique. En effet, la déflectométrie possède plusieurs variantes, chacune avec leur propre intérêt. On distingue cinq grandes catégories : Hartmann inversé, Phase Measuring Deflectometry, Direct Phase Measuring Deflectometry, Stereo Deflectometry et Modal Phase Measuring Deflectometry.

**Le chapitre 2** évoque la mise en pratique d'une technique de déflectométrie. En effet, le système est confronté à plusieurs erreurs systématiques diminuant la précision des mesures. On retrouve des erreurs liées à la quantification du motif affiché sur l'écran pour sonder la surface, à la distorsion causée par les lentilles de la caméra, à la non-linéarité dans l'affichage des intensités sur un écran,

etc. Leur connaissance est cruciale et nécessite une calibration du système pour réduire, voire de supprimer, leurs effets sur les mesures. Ce chapitre explique principalement comment faire face à la non-linéarité d'un écran.

Etant donné que la crise pandémique du COVID-19 a provoqué la fermeture de tous les laboratoires, il est devenu impossible de poursuivre l'étude de ce mémoire avec le dispositif élaboré au Centre Spatial de Liège. Dès lors, nous avons réorienté le travail vers une étude entièrement numérique couvert dans **le chapitre 3**. Un code de ray tracing a été élaboré sur Matlab se rapprochant au maximum du montage du chapitre 2. Par la suite, la prise de données et les algorithmes de traitement ont été mis au point pour retirer les informations voulues. Dans ce chapitre, le problème du désalignement de miroir est abordé de la façon suivante : l'écran affiche un motif en niveaux de gris dont l'amplitude est modulée par une fonction cosinus donnant des franges horizontales et verticales. L'idée est de cartographier le miroir au moyen de la phase. La caractérisation des cartes de phase s'effectue au moyen des coefficients des polynômes de Zernike. Le but ultime est de caractériser un désalignement inconnu en se basant uniquement sur le comportement de ces coefficients.

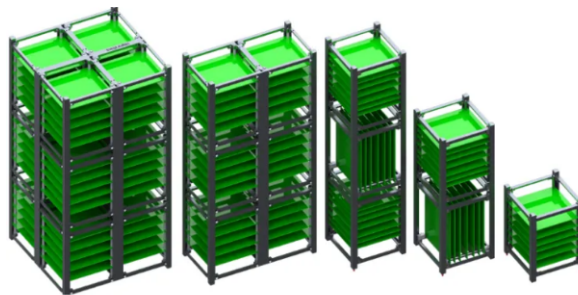
Ce mémoire se termine avec la conclusion de tout le travail de cette année au **chapitre 4**. Diverses pistes de réflexions sont également évoquées pour continuer cette étude.

# Chapitre 1

## Notions préliminaires et état de l'art

### 1.1 Les CubeSats

Le standard CubeSat est un concept développé pour la première fois par l'Université polytechnique de Californie et par le laboratoire de développement des systèmes spatiaux de l'Université de Stanford en 1999 [1]. Ceux-ci ont mis en place ce standard afin de rendre le spatial plus accessible en réduisant les coûts de fabrication et de lancement. Les universités et les firmes privées peuvent dès lors créer et placer en orbite leur propre satellite [2]. De plus, ces satellites n'ont plus besoin d'être pensés en fonction de l'interface du lanceur et leur mise en oeuvre est relativement rapide (2 à 3 ans) [3]. Le premier CubeSat fut lancé par une fusée russe le 30 juin 2003. Le CubeSat est défini comme étant un satellite miniature en forme de cube de 10 cm d'arête et dont la masse maximale est de 1,33 kg. Cette convention est appelée 1U [4][5]. Pour tester des charges utiles beaucoup plus importantes, plusieurs standards 1U sont assemblés [2] donnant lieu à des satellites dits 2U, 3U, etc, comme le montre la figure 1.1.



**FIGURE 1.1** – Il existe plusieurs formats de CubeSat. Ceux-ci sont obtenus par assemblage d'une succession de standards 1U. De gauche à droite : 12U, 6U, 3U, 2U et 1U. [2]

*CubeSat Design Specification* [6] défini par l'Université de Californie est un cahier de charge qui donne les spécifications requises pour la création des CubeSats. Citons quelques éléments :

- Le centre de gravité du CubeSat doit être positionné de manière très précise en fonction du nombre d'unités. Par exemple, un CubeSat 1U possède son centre de gravité à moins de 2 cm du centre géométrique sur l'axe Z.
- Aucun élément ne doit être libéré dans l'espace. De plus, les CubeSats se désintègrent totalement dans l'atmosphère en fin de mission.

- Seul le CubeSat limite la taille des éléments déployables. Ces derniers ne sont pas tenus par le P-POD<sup>1</sup>.
- Plusieurs conditions existent sur les rails : largeur de 8,5 mm, rugosité à 1,6  $\mu\text{m}$ , ils doivent être anodisés pour l'isolation électrique, etc.
- La pression interne sur les composants ne doit pas dépasser 1,2 atm.

Les CubeSats sont essentiellement conçus pour des démonstrations technologiques miniaturisées (capteurs, systèmes de contrôle d'attitude, etc) et pour des applications terrestres (localisations de véhicules, mesures de paramètres environnementaux, etc)[2].

Pour ne citer que deux exemples :

- OUFTI-1 (Orbital Utility For Telecommunication) [3] [7] est le premier CubeSat belge réalisé par des étudiants de l'Université de Liège, de l'institut de Gramme et de l'ISIL avec la supervision du CSL, SPACEBEL et TAS ETCA. Son lancement a eu lieu le 25 avril 2016 sur la base de Kourou. Son but est de tester le protocole de radiocommunication D-STAR (Digital Smart Technology for Amateur Radio) dans l'espace afin d'améliorer la communication dans les zones isolées ou désastrées. Sa représentation est montrée sur la figure 1.2.

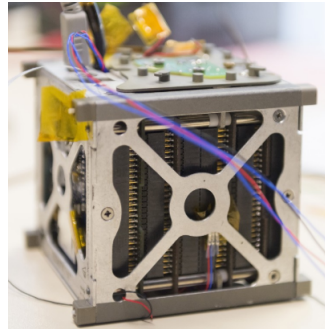


FIGURE 1.2 – CubeSat 1U OUFTI-1. [7]

- Eye-Sat [8] [9] est un CubeSat de type 3U dont la mission est d'étudier la lumière zodiacale<sup>2</sup> et la voie lactée, mais aussi de tester des technologies matures de R&T CNES de Toulouse. Ce CubeSat s'inscrit dans le projet JANUS (Jeunes en Apprentissage pour la réalisation de Nanosatellites des Universités et des écoles de l'enseignement Supérieur). Il a été lancé le 18 décembre 2019 à Kourou. Sa représentation est donnée à la figure 1.3.

L'élaboration d'un système de déflectométrie doit tenir compte de toutes les restrictions qu'impliquent un CubeSat. En effet, le cahier des charges impose des conditions sur la taille, la composition, etc. d'un CubeSat. La rigueur de fabrication du nanosatellite conditionne la mise en pratique d'un système de métrologie. Ce mémoire se concentrera surtout sur l'aspect miniaturisable des éléments afin de ne pas accéder le volume requis.

1. Il s'agit de l'interface entre le lanceur et le CubeSat. Du nom de *Poly Picosatellite Orbital Deployer*, cette structure rectangulaire en aluminium anodisé peut contenir jusqu'à 3U. Les CubeSats y sont éjectés par un système sur rail et à ressort. [3]

2. Lumière émise suite à la réflexion de la lumière solaire sur la poussière interstellaire. [2]

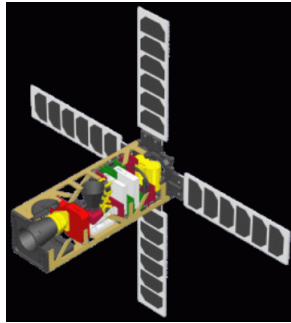


FIGURE 1.3 – CubeSat 3U Eye-Sat. [8]

## 1.2 Structures déployables

Ce mémoire a pour but de développer un système de métrologie optique afin de vérifier l’alignement de miroirs sur un satellite de type CubeSat. Cependant, l’étude ne doit pas se focaliser uniquement sur cet objectif. En effet, il ne s’agit pas seulement de savoir si le miroir est aligné ou non, mais il est aussi indispensable de se renseigner s’il existe un moyen de le repositionner dans la précision voulue. Cette section passera donc en revue de manière non exhaustive différents systèmes de déploiement. Cet état de l’art est indispensable afin de cerner la possibilité d’une mise en pratique d’une technique déflectométrique.

Une structure déployable est une structure qui change de configuration réversiblement ou non [10]. Le but de ces structures est d’économiser grandement de la place lorsque l’objet n’est pas déployé, facilitant notamment son transport. Avant de les retrouver dans l’industrie spatiale, elles sont surtout issues de notre merveilleuse nature [11] comme l’illustre la figure 1.4.



FIGURE 1.4 – Exemples de déploiement dans la nature. Le déploiement de pétales de fleurs en haut et d’ailes d’insectes en bas. La structure déployable est dans son état replié quand sa fonction n’est pas sollicitée. [11]

Comme mentionné précédemment, les structures déployables sont très intéressantes pour le spatial. En effet, étant donné que les satellites sont de plus en plus grands et que le volume disponible dans les lanceurs<sup>3</sup> est limité, une partie du volume du satellite doit être réduite grâce à cette technologie [11].

Dans l’espace, il est possible de déployer des antennes, des panneaux solaires, des voiles solaires, des membranes, etc [5]. Ce mémoire se concentrera sur des miroirs déployables. Par conséquent, cer-

3. Par exemple, la coiffe d’Ariane 5 possède une hauteur de 17 m pour un diamètre de 4,57 m. [11]

taines structures déployables peuvent directement être mises à l'écart car elles ne sont pas adaptées au déploiement de miroirs. Citons quelques exemples de dispositifs à proscrire.

### Dispositifs à proscrire

Un premier système de déploiement sont les structures articulées [5]. Il s'agit d'éléments rigides reliés par des articulations. Elles peuvent prendre diverses formes mais la plus connue est le pantographe. Ces structures sont relativement simples à mettre en oeuvre. De plus, elles possèdent une bonne rigidité, résistant bien aux efforts mécaniques présents dans un milieu spatial. Cependant, la précision lors du déploiement est assez mauvaise et nécessite un système de correction [11]. De plus, il existe un risque non négligeable que les structures soient polluées à cause du lubrifiant des articulations. Les pantographes sont notamment utilisés sur le CubeSat Falcon SAT-7 comme le montre la figure 1.5. Le déploiement s'effectue avec des pantographes à ressorts qui une fois déployés donnent de la tension à une membrane pour récolter des photons. Cet exemple est un déploiement passif,

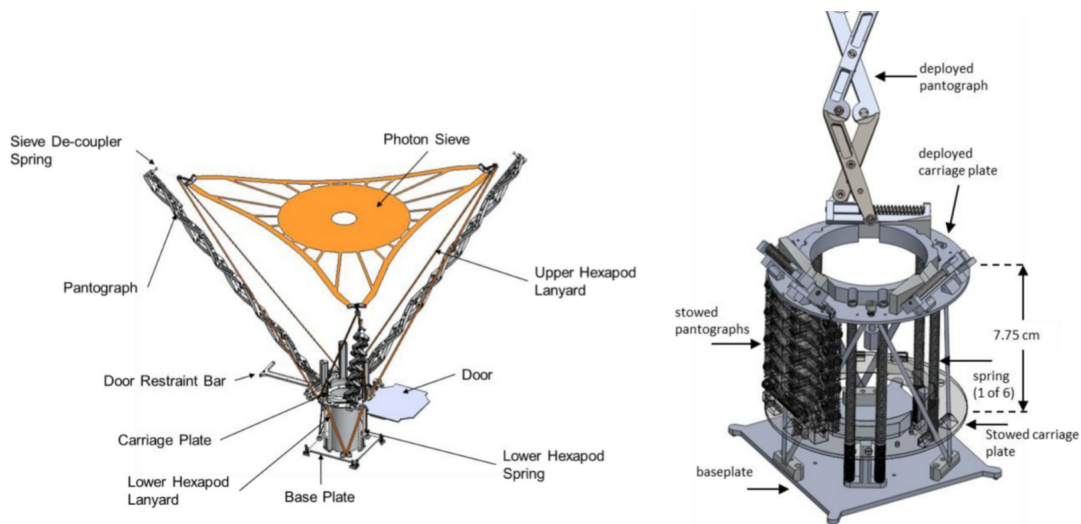


FIGURE 1.5 – CubeSat 3U Falcon SAT-7 et sa structure déployable de type pantographe. [12]

c'est-à-dire basé sur l'énergie de déformation élastique. Il existe d'autres configurations où le déploiement est actif c'est-à-dire nécessitant un moteur [10]. C'est le cas de l'antenne rigide nommée SSDA (*Solid Surface Deployable Antenna*) illustré sur la figure 1.6. Cette première méthode de déploie-

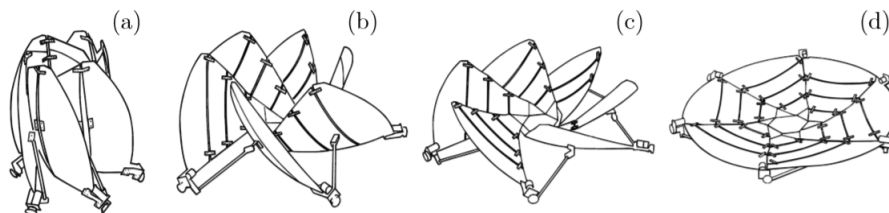
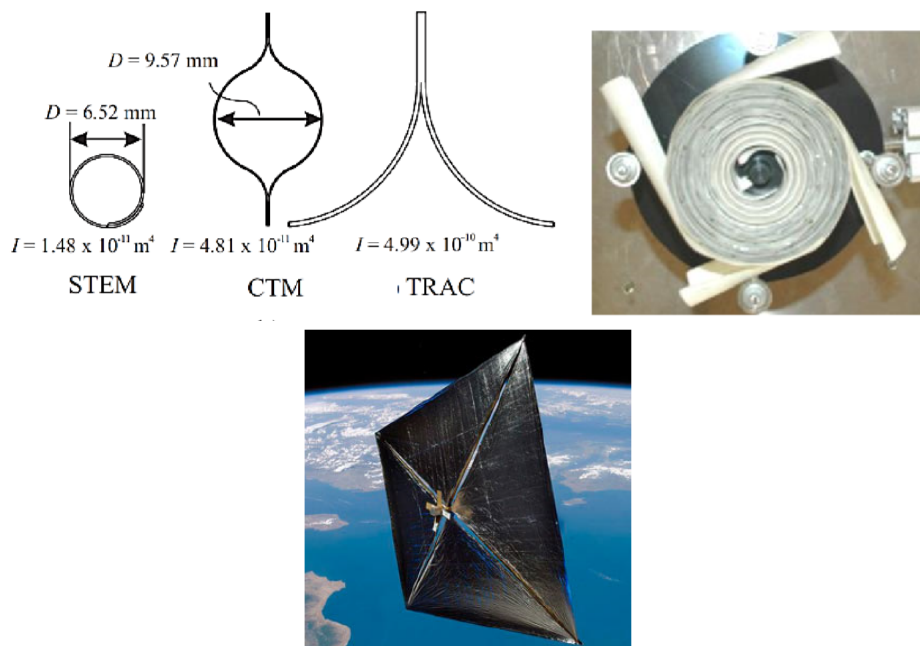


FIGURE 1.6 – Etapes du déploiement actif de SSDA (*Solid Surface Deployable Antenna*). [10]

ment ne convient pas au contexte étudié pour ce mémoire pour plusieurs raisons. Dans un premier

temps, il s'agit de structures assez lourdes ne convenant pas directement pour une application sur les CubeSats [5]. Le seul cas possible est si l'objet à déployer ne dispose pas d'une masse conséquente comme la membrane de Falcon SAT-7. En ce qui concerne les miroirs, plusieurs pantographes par miroir seront nécessaires augmentant considérablement la masse du CubeSat. Deuxièmement, aucun dispositif de correction n'est mis en oeuvre. La configuration atteinte en fin de processus de déploiement ne change pas.

Un autre type de technologie déployable est le mât enroulé [5]. Il s'agit d'une structure flexible dont le déploiement est passif. Ces tubes possèdent une double configuration : plate à l'état replié et en forme de tube à l'état déplié. Ces longs mâts de sections variées sont enroulés autour d'un moyeu et stockent l'énergie de déformation Il permettent de déployer des structures légères comme des voiles solaires ainsi que l'illustre la figure 1.7.



**FIGURE 1.7** – En haut à gauche : les différentes formes de mâts enroulables (STEM, CTM et TRAC) [13]. En haut à droite : mâts enroulés autour du moyeu [13]. En bas : CubeSat NanoSail-D disposant de cette technologie pour sa voile solaire [5].

Un troisième type de structure sont les systèmes gonflables [11]. Dans cette catégorie, on retrouve les réflecteurs gonflables qui sont extrêmement légers et d'une grande compacité. Cependant, la flexibilité du système rend le déploiement trop peu précis. Le principe est simple : un gaz gonfle la structure et le soleil solidifie la membrane de par sa chaleur rayonnée. Un exemple pour cette classe est l'antenne ISRS (*Inflatable Space Rigidised Structure*) présentée à la figure 1.8.

Pour des informations complémentaires, en plus des articles mentionnés, sur les structures déployables, la review [14] en propose une classification.

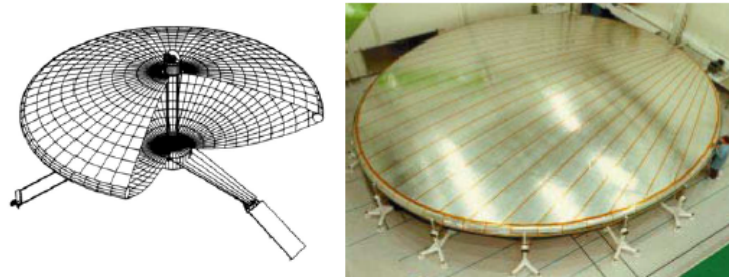


FIGURE 1.8 – L’antenne gonflable ISRS (*Inflatable Space Rigidised Structure*). [11]

### Dispositif à envisager

Pour trouver un type de déploiement convenant au miroir, il faut se concentrer sur les mécanismes d’alignement. L’alignement des miroirs fait entrer en jeu 3 paramètres mécaniques du déploiement : le piston, le tip et le tilt [15]. La figure 1.9 illustre l’effet géométrique de ces 3 paramètres. Une description plus précise d’un point de vue mathématique sera effectuée dans le chapitre 3 sur la partie Polynômes de Zernike. Il faut donc trouver un système de déploiement pouvant faire entrer en jeu ces trois paramètres.

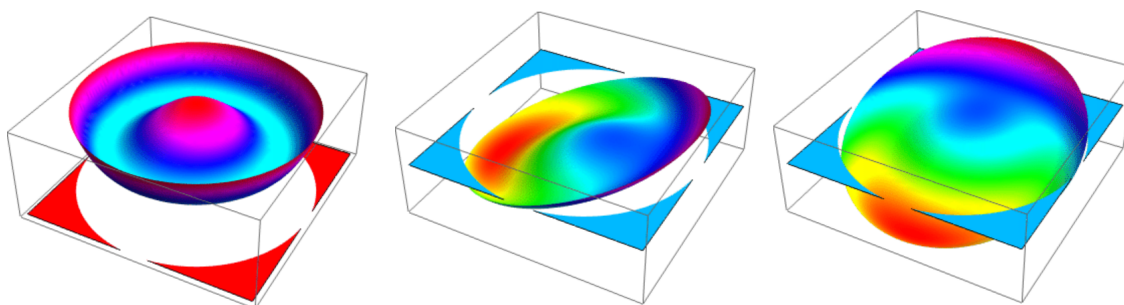
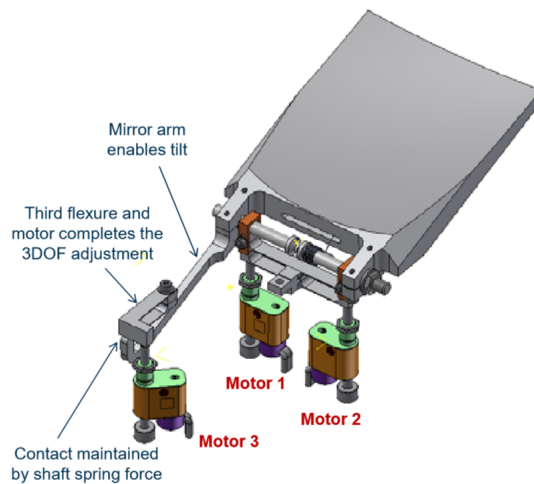


FIGURE 1.9 – De gauche à droite : piston, tip et tilt d’une surface par rapport à une surface de référence. [16]

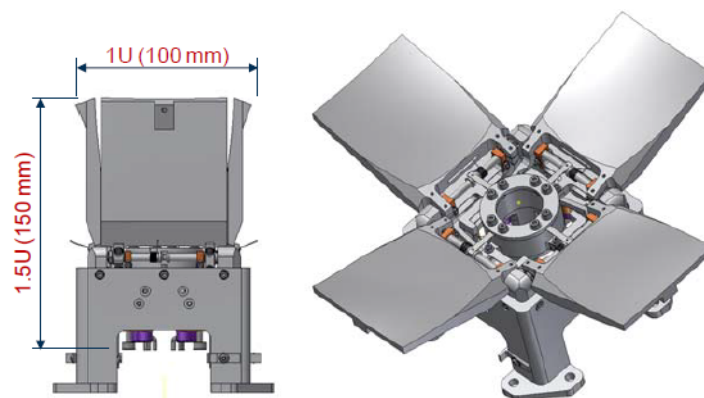
Suite à une analyse de la littérature, une méthode de déploiement peut être adaptée à notre étude. Il s’agit de miroirs déployables à l’aide de ressorts de torsion [17]. A la fin du déploiement, les miroirs se trouvent sur des barres de flexion dont l’utilité est d’ajuster l’alignement. En effet, ces barres en aluminium sont en contact avec 3 moteurs piézoélectriques qui permettent de bouger le miroir en piston, tip et tilt. La mécanique est illustré à la figure 1.10. L’avantage de ce système est donc la possibilité de corriger le déploiement, mais aussi de garder la position lorsque les moteurs sont éteints.

Ce mémoire traite du déploiement de miroirs. Le design de ceux-ci est avant tout fortement impacté par le mode de pliage et la technique de déploiement. Afin de travailler dans un contexte le plus réaliste possible, nous serons amenés à travailler sur un prototype ayant la même configuration que l’exemple de la figure 1.11.

La connaissance de l’environnement dans lequel un système de déflectométrie doit être mis en oeuvre est importante. En effet, les recherches effectuées sur les CubeSats et sur les systèmes déployables montrent que l’élaboration d’un dispositif de déflectométrie est possible dans ce contexte.



**FIGURE 1.10** – Miroir avec sa structure déployable. Celle-ci est en contact avec trois moteurs pour aligner le miroir en piston, tip et tilt. [17]



**FIGURE 1.11** – Design d'un CubeSat. A gauche : forme repliée. A droite : forme dépliée. [17]

Ces deux sections donnent donc du sens pour la suite de ce travail.

### 1.3 Métrologie optique

La métrologie optique est indispensable pour tester la qualité des surfaces en utilisant les propriétés de la lumière. Ce moyen de mesure fournit des résultats rapides grâce à la vitesse de propagation de la lumière. De plus, elle présente aussi l'avantage d'être sans contact. Le but de cette section est d'introduire la notion de déflectométrie et de passer en revue ses différents développements. L'idée est de prendre connaissance de ce qui se fait dans la littérature de manière générale tout en développant plus en profondeur les points intéressants pour ce mémoire.

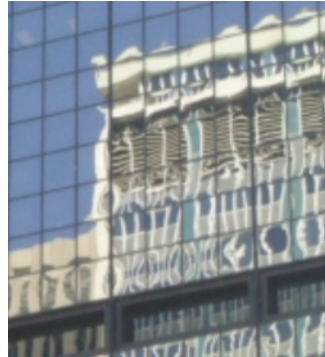
### 1.3.1 Méthodes Interférométriques

Suivant le type d'interaction que la lumière possède avec la surface sous test, les méthodes interférométriques peuvent être mises en oeuvre de différentes manières [18]. On parle de techniques basées sur *la diffraction* et basées sur *la diffusion*. Cependant, l'idée principale reste la même. Le but est d'obtenir la phase  $\psi(t) = \omega t + \phi$  d'une onde électromagnétique en utilisant la superposition de signal [19]. En effet, la fréquence de l'onde visible est trop élevée (de l'ordre de  $10^{14} Hz$ ) pour pouvoir enregistrer les variations temporelles de la phase. L'interférence est donc mise en place pour pouvoir s'affranchir du terme  $\omega t$ .

### 1.3.2 Méthode non interférométrique : la déflectométrie

Le principe fondamental de la déflectométrie [18] est basé sur une des lois les plus connues en optique : la loi de la réflexion.

Inconsciemment, chacun a déjà expérimenté dans la rue l'essence même de cette technique. En effet, la figure 1.12 illustre la réflexion d'un bâtiment sur un immeuble vitré. Cette image réfléchie est déformée à cause des défauts de surface, comme les défauts de planéité des fenêtres. La déformation est facile à expliquer en considérant le trajet de la lumière. Suivant la loi de la réflexion, l'angle incident est égal à l'angle réfléchi par rapport à la normale au point d'incidence de la surface. Le déplacement de la normale entraîne donc un déplacement du rayon réfléchi perçu par l'oeil. La variation de la direction de la normale peut être assimilée à la variation de la pente locale, également appelé la courbure locale de la surface. Cependant, notre oeil est incapable de quantifier cette variation, d'où la mise en place d'un système de métrologie appelé la déflectométrie.

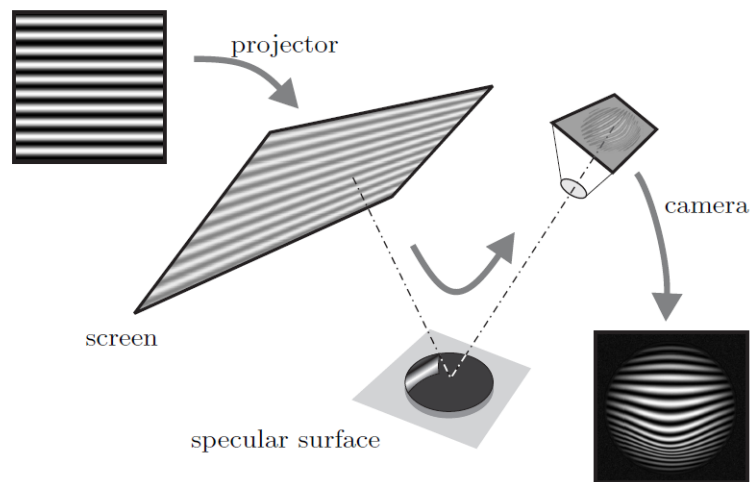


**FIGURE 1.12** – Réflexion d'un bâtiment sur un immeuble vitré. La distorsion de l'image est provoquée par des défauts présents dans le vitrage. Le déplacement de la normale à la surface suite aux défauts entraîne une variation des rayons réfléchis suivant la loi de la réflexion. [18]

Comme mentionné dans l'exemple, l'analyse de la lumière réfléchie par une surface spéculaire<sup>4</sup> permet de remonter aux pentes locales de cette surface sous test [20]. La topologie de la surface est reconstruite grâce à diverses techniques, notamment l'intégration du gradient de pente.

Concrètement, la déflectométrie est réalisée à l'aide d'une caméra, d'une surface test et d'un moniteur (un écran type LCD par exemple) comme illustré sur la figure 1.13. Ce dernier est assimilé comme une source lumineuse dans le dispositif lorsqu'un motif y est affiché. Le motif doit être une

4. Un objet spéculaire est un objet qui réfléchit un faisceau lumineux dans une seule direction, contrairement à un objet diffus. [20]



**FIGURE 1.13** – Schéma du principe fondamental de la déflectométrie. Le motif périodique sur un écran est réfléchi par la surface vers la caméra. L'image capturée est déformée à cause de la forme de l'objet spéculaire. [20]

figure facilement analysable et, de préférence, périodique<sup>5</sup>. Dans la littérature, les principaux motifs utilisés sont binaires ou sinusoïdaux [21]. Ensuite, cette lumière structurée est envoyée sur un objet spéculaire dont la surface doit être mesurée. Chaque rayon lumineux est donc réfléchi dans une seule direction déterminée par la loi de la réflexion. À noter qu'au vu de la nature spéculaire de la surface, la taille du moniteur est primordiale afin de couvrir l'ensemble de la surface sous test [20]. La caméra capture l'image réfléchie qui va être analysée à l'aide d'algorithmes. La topologie de la surface est déterminée en calculant les variations des pentes locales et en passant, par exemple, par des méthodes d'intégration. Ces mesures doivent être effectuées dans toutes les directions et nécessitent donc une lumière avec des motifs orthogonaux. Il existe plusieurs moyens d'accéder aux pentes locales et de reconstruire la surface par la déflectométrie. Ceux-ci sont présentés dans la suite de ce chapitre.

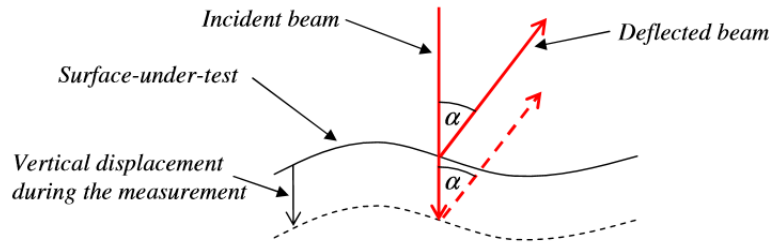
### Sensibilité par rapport aux variations de pentes

On parle de déflectométrie 3D lorsque la métrologie utilise la réflexion des franges pour reconstruire une surface sur base des pentes locales. Un avantage par rapport à une mesure directe de la hauteur est que la valeur de la pente n'est pas influencée par la translation dans la direction d'observation de la surface [22], ce qui rend cette déflectométrie insensible aux vibrations. En effet, la notion de pente ne contient pas d'informations sur la hauteur absolue de l'objet comme illustré sur la figure 1.14.

Du fait de cette grande caractéristique, la déflectométrie est largement utilisée en industrie afin de contrôler au nanomètre près la forme de la surface fabriquée.

La technique de déflectométrie de base permet de mesurer différentes formes de surface : sphériques, paraboliques, de formes libres, etc. Cependant, il est impossible de reconstruire des objets

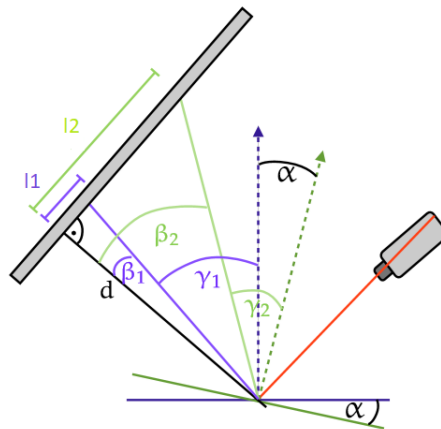
5. Le motif périodique est privilégié par rapport au motif aléatoire car le bruit dans la périodicité peut facilement être analysé. Un motif aléatoire n'est rien d'autre qu'un bruit qui est difficilement analysable. De plus, la périodicité d'un motif permet de quantifier la surface, c'est-à-dire associer un nombre bien précis à un endroit de la surface. La résolution s'en retrouve dès lors meilleure. Cependant, un gros désavantage d'un codage périodique est la répétition du motif provoquant des ambiguïtés (cette partie sera détaillée dans les prochaines sections). [18]



**FIGURE 1.14** – Mesure déflectométrique d’une surface. L’angle de la pente  $\alpha$  ne varie pas suite à un déplacement vertical de l’objet. [22]

qui présentent des marches. En effet, ce genre de configuration peut être vue comme une succession de petites surfaces décalées en hauteur. La mesure des marches n’est pas faisable en utilisant la déflectométrie de base. Pour remédier à ce problème, des améliorations dans la mise en oeuvre seront apportées à la déflectométrie donnant la Direct Phase Measuring Deflectometry décrite dans la suite de ce chapitre.

La déflectométrie permet d’étudier des surfaces en passant par une analyse de pentes. Il est intéressant d’étudier la sensibilité de cette technique par rapport aux variations de la pente. Pour mieux comprendre, la figure 1.15 illustre la situation qui va être développée ci-après . Afin d’analyser la



**FIGURE 1.15** – Sensibilité d’un dispositif de déflectométrie par rapport à un changement de pente. La surface passe de la configuration mauve à la verte en étant inclinée d’un angle  $\alpha$ . [23]

sensibilité, la configuration passe de la situation en bleu vers la situation en vert en inclinant la surface sous test d’un angle  $\alpha$  [23]. La question est de combien de pixels le rayon va t-il changer d’une configuration à l’autre. Il s’agit donc d’obtenir une expression de  $l_2 - l_1$ . L’idée de cette démonstration est de relier  $l_2 - l_1$  à l’angle  $\alpha$ . Pour ce faire, il faut passer par une analyse des relations des angles. Deux relations sont rapidement mis en évidences.[23]

$$\gamma_1 - \gamma_2 = \alpha \quad (1.1)$$

$$\beta_2 - \beta_1 = \gamma_1 - (\gamma_2 - \alpha) = 2\alpha \quad (1.2)$$

La dernière égalité de (1.2) est obtenue en remplaçant avec (1.1). Ensuite, la variation de la position du rayon peut s'exprimer comme [23]

$$\begin{aligned}
 \Delta l &= l_2 - l_1 \\
 &= d \tan(\beta_2) - d \tan(\beta_1) \\
 &= d \tan(\beta_2 - \beta_1)(1 + \tan(\beta_1) \tan(\beta_2)) \\
 &= d \tan(2\alpha)(1 + \frac{l_1 l_2}{d^2}) \\
 &= d \tan(2\alpha)
 \end{aligned}
 \tag{1.3}$$

Tout le développement est basé sur des relations trigonométriques. La dernière expression de (1.3) est obtenue en considérant  $l_1 = 0$ . Cette approximation est souvent réalisée dans la littérature. Cependant, elle n'est valable que pour certains points de la surface permettant néanmoins une étude sur la limite théorique inférieure [23] [20].

### Limites physiques

Il est possible d'étudier les limites physiques de la technique de déflectométrie. Cependant, pour y parvenir, plusieurs approximations doivent être appliquées. Les approximations sont les suivantes [23]

- L'espace n'est pas discrétisé avec la présence de pixel.
- La surface sous test peut localement être assimilée à une parabole.
- L'axe de parabole est le lieux de convergence des rayons lumineux.
- Les rayons lumineux arrivant à la surface quittent perpendiculairement l'écran.
- Les limites physiques du problème s'expriment sous la forme d'une relation d'incertitude. Ainsi, il doit exister une relation entre l'incertitude sur la pente,  $\delta\alpha$ , et l'incertitude de la position mesurée sur le miroir,  $\delta x$ .

Le système étudié est illustré à la figure 1.16. Pour obtenir une relation d'incertitude, il faut d'abord

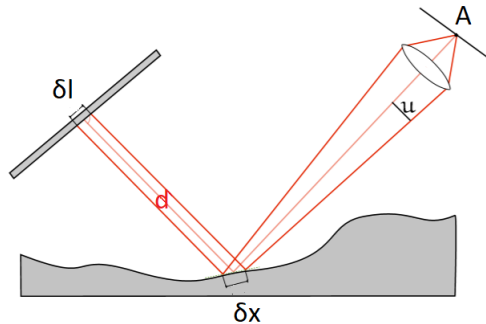


FIGURE 1.16 – Schéma du principe de déflectométrie avec les approximations mises en place. [23]

partir sur la notion de limite de diffraction. Ainsi, un pixel de la caméra voit sur la surface une certaine région  $\delta x$  définie comme [20]

$$\delta x = \frac{\lambda}{\sin u}
 \tag{1.4}$$

Où  $\delta x$  est aussi appelé la résolution latérale,  $\lambda$  est la longueur d'onde de la lumière utilisée et  $u$  est le demi angle d'ouverture de la caméra.

La région de l'écran  $\delta l$  projetée sur la surface en  $\delta x$  peut être approximée par [20]

$$\delta l \approx 2d \tan u \approx 2d \sin u = 2d \frac{\lambda}{\delta x} \quad (1.5)$$

Cette approximation n'a de sens que si la distance  $d$  entre l'écran et la surface est plus grande que la profondeur de mise au point. De plus, l'angle d'ouverture  $u$  est supposé assez petit pour remplacer la fonction tangente par la fonction sinus. Ainsi, la dernière étape est obtenue en remplaçant avec l'expression (1.4).

A ce stade du développement, une expression de  $\delta x$  est obtenue sur base du critère de Rayleigh [23]. Afin de trouver le principe d'incertitude reliant  $\delta x$  à  $\delta \alpha$ , une expression de  $\delta \alpha$  doit être mise en place. Pour ce faire, cette incertitude angulaire peut être reliée à l'incertitude de la phase mesurée  $\delta \varphi$  [20].

$$\delta \varphi = 2d \tan \delta \alpha \quad (1.6)$$

Dans le cas d'un déphasage en plusieurs étapes, Lampalzer ([24] pour avoir plus de détails) a prouvé que l'incertitude sur la phase dépendait de 3 paramètres [20].

$$\delta \varphi = \frac{p}{QC} \quad (1.7)$$

Où  $p$  est la période du signal.  $Q$  est une constante appelée le facteur de qualité. Ce facteur tient compte de l'impact du bruit de la numérisation des images par la caméra.  $C$  est le contraste du motif utilisé.[20]

$$C = \frac{\sin(\pi \delta l / p)}{\pi \delta l / p} \quad (1.8)$$

L'incertitude sur la phase est minimale lorsque le sinus est maximal, c'est-à-dire quand il vaut 1. Dès lors, la période  $p$  du motif doit être égale à  $2\delta l$ . Ainsi, en remplaçant (1.7) avec (1.4), la valeur minimale de  $\delta \varphi$  est donnée par[20]

$$\delta \varphi \geq \frac{2\pi \lambda d}{Q \delta x} \quad (1.9)$$

De part cette expression, le principe d'incertitude peut enfin être mise en place. En effet, en utilisant les deux expressions de  $\delta \varphi$ , c'est-à-dire les équations (1.6) et (1.9), on obtient [23]

$$\delta \alpha \cdot \delta x \geq \frac{\pi \lambda}{Q} \quad (1.10)$$

Ce principe d'incertitude donne la limite inférieure aux mesures déflectométriques. Au vu des approximations réalisées, cette limite n'est que théorique car elle fournira une estimation plus faible que la réalité. Cependant, elle reste intéressante pour fournir un ordre de grandeur dans les mesures.

Les avantages de la déflectométrie sont divers. Il s'agit avant tout d'une technique sans contact et non destructive qui permet de mesurer des défauts locaux [18]. Cette mesure est rapide (quelques secondes), robuste et précise [25]. Les dispositifs permettant sa réalisation sont peu coûteux et très flexibles. Une brève comparaison entre la déflectométrie et l'interférométrie permet de mettre en lumière les raisons de choisir la déflectométrie pour ce mémoire [19] :

- Elle présente une insensibilité aux vibrations rendant son utilisation adaptée en milieux industriels. L'interférométrie est très sensible aux vibrations et aux changements environnementaux (pression, température).

- La déflectométrie est facile à mettre en oeuvre contrairement à l'interférométrie. Cette dernière est très coûteuse et pas du tout flexible. En effet, elle nécessite une optique de compensation pour l'étude de formes de surface plus complexes.
- La sensibilité de la mesure pour la déflectométrie est purement géométrique tandis que l'interférométrie est conditionnée par la longueur d'onde de la lumière utilisée.
- Un des plus gros désavantages de la déflectométrie est qu'elle n'est pas aussi précise que l'interférométrie. Cependant, dans le domaine des submicromètres, elle peut sérieusement concurrencer l'interférométrie grâce aux nouvelles calibrations apportées au système.

### Hartmann inversé

Dans un premier temps, les pentes locales peuvent être calculées par le modèle d'Hartmann inversé [26] (cf. Fig. 1.17).

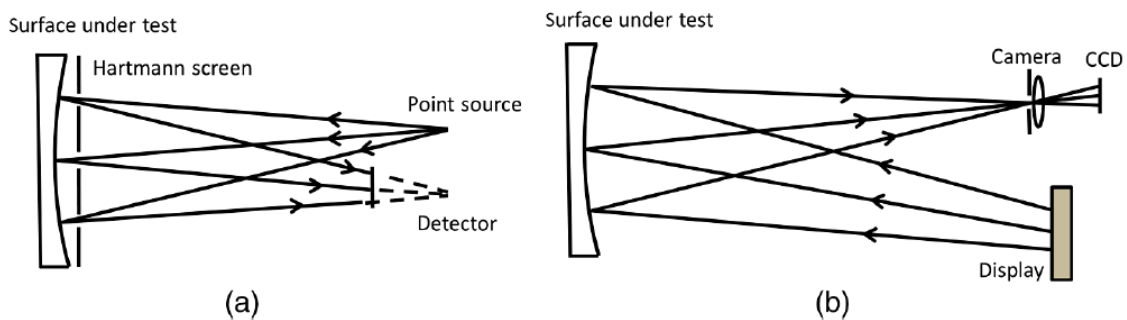


FIGURE 1.17 – Géométrie du test Hartmann (a) et du test Hartmann inversé (b). [27]

La compréhension du mécanisme du test d'Hartmann inversé est plus aisée en introduisant avant tout le test d'Hartmann [25]. Dans ce test, la source lumineuse ponctuelle est placée au centre de courbure et le détecteur près du foyer. Un écran troué formant un quadrillage dit *écran Hartmann* est placé devant la pupille du miroir à tester. Son effet est d'échantillonner le miroir. Seuls les rayons passant par les trous sont réfléchis sur le détecteur. La position des pixels illuminés du détecteur renseigne sur la pente de la surface au niveau des trous du masque. Les erreurs de surface peuvent être donc détectées en comparant les pentes avec celle d'une surface parfaite.

Dans le test d'Hartmann inversé, tous les rôles sont inversés [28] [29] : le sens des rayons, le détecteur devient l'écran affichant un motif lumineux codé et la source ponctuelle est remplacée par la caméra centrée sur le miroir. Le motif lumineux est souvent codé en binaire. Par exemple, un seul pixel est éclairé en blanc et le reste de l'écran est noir. La caméra voit donc une zone qui brille sur le miroir, appelée *pixel miroir*. Cette région signifie que les angles d'incidence et de réflexion sont égaux par rapport à la normale de la surface (qui n'est rien d'autre que la bissectrice angulaire). La caméra peut donc être vue comme l'écran Hartmann qui échantillonne la surface grâce à ses pixels éclairés. La position des pixels caméra éclairés donne une information sur la pente locale. Avec une triangulation entre les pixels écran, pixels miroir et position pixels caméra, la pente locale peut être calculée à l'aide des équations suivantes [28] :

$$w_x(x_m, y_m) = \frac{\frac{x_m - x_s}{d_{m2s}} + \frac{x_m - x_c}{d_{m2c}}}{\frac{z_{m2s} - W(x_m, y_m)}{d_{m2s}} + \frac{z_{m2c} - W(x_m, y_m)}{d_{m2c}}} \quad (1.11)$$

$$w_y(x_m, y_m) = \frac{\frac{y_m - y_s}{d_{m2s}} + \frac{y_m - y_c}{d_{m2c}}}{\frac{z_{m2s} - W(x_m, y_m)}{d_{m2s}} + \frac{z_{m2c} - W(x_m, y_m)}{d_{m2c}}} \quad (1.12)$$

Où  $w_x$  et  $w_y$  sont les pentes locales en  $x$  et en  $y$  aux points de coordonnées  $(x_m, y_m)$ ;  $x_m$  et  $y_m$  sont les coordonnées du pixel miroir;  $x_c$  et  $y_c$  sont les coordonnées de l'ouverture de la caméra;  $x_s$  et  $y_s$  sont des coordonnées du pixel de l'écran affichant le motif structuré;  $z_{m2s}$  (respectivement  $z_{m2c}$ ) est la différence de coordonnées  $Z$  entre le miroir et l'écran (la caméra respectivement);  $d_{m2s}$  (respectivement  $d_{m2c}$ ) est la distance entre le miroir et l'écran (la caméra respectivement) et  $W(x_m, y_m)$  est la profondeur de la surface aux points  $(x_m, y_m)$ . La figure 1.18 illustre la signification de ces différents termes.

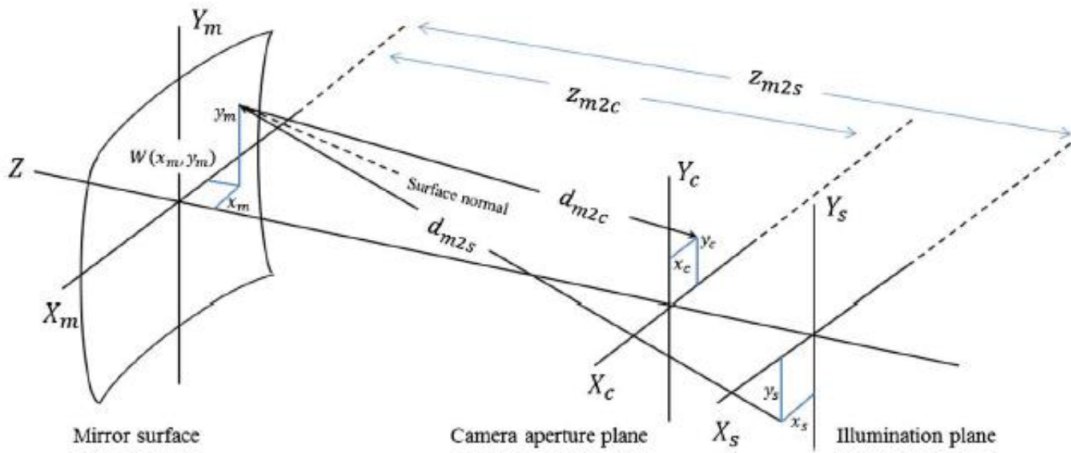


FIGURE 1.18 – Géométrie du calcul de pentes dans le test Hartmann inversé. [26]

La surface  $w$  d'un miroir est obtenue par intégration des pentes. Cependant, un pixel miroir peut être éclairé par plusieurs pixels écran à cause de la taille finie de l'ouverture de la caméra. Dès lors, le calcul de la pente au niveau du pixel miroir se fait par la méthode centroïde sur les pixels écran.

Cette méthode, contrairement à la PMD décrite plus tard, n'utilise pas les informations de phase de la fonction générée [21]. Le motif est codé en binaire soit un éclairant pixel par pixel en blanc avec le reste de l'écran noir, soit en éclairant des lignes de pixels ou encore des motifs carrés. Le balayage carré a pour avantage de posséder le plus grand SNR par rapport aux deux autres méthodes de balayage. L'application de la méthode centroïde dans cette technique de déflectométrie consiste à évaluer le centre de tous les pixels écran éclairant le pixel miroir. En d'autres mots, il s'agit de faire une moyenne pondérée avec les intensités des pixels écran [21].

$$x_{screen} = \frac{\sum_{i \in ESP} x_i I_i}{\sum_{i \in ESP} I_i} \quad (1.13)$$

### Discussion

La méthode centroïde est sensible aux variations spatiales de la lumière et non aux variations temporelles. En effet, l'intensité sur une même image change en fonction du pixel. De plus, en général, le SNR est très bon car chaque pixel est balayé par une intensité maximale. Pour ce mémoire, la configuration Hartmann inversé est particulièrement intéressante de par sa simplicité de mise en oeuvre. Un nombre restreint d'éléments est une bonne chose pour envisager la miniaturisation du système. Cependant, le motif utilisé ne semble pas approprié pour le contexte de ce mémoire. En effet, son utilisation permet de trouver directement la pente locale au moyen des coordonnées de chaque composant. Dans ce mémoire, l'objectif est justement de connaître la position du miroir. Le motif binaire peut donc être proscrit.

### **Phase Measuring Deflectometry**

L'évolution de la technique de déflectométrie se fait en améliorant notamment l'affichage et l'observation du motif à analyser. Pour accélérer le processus de détermination de la forme globale de la surface, la méthode PMD (Phase Measuring Deflectometry) scanne des motifs 2D orthogonaux, à savoir des franges sinusoïdales horizontales et verticales, avec plusieurs déphasages. Outre sa rapidité, la méthode PMD présente aussi l'avantage d'être d'une grande précision dans les mesures. De plus, elle fonctionne plein champ et le traitement des données est automatique [30].

Le principe de base [31] repose sur la correspondance en phase entre un pixel écran et un pixel caméra. L'intensité des pixels écran varie suivant une fonction sinusoïdale. Celle-ci sera déphasée plusieurs fois d'une certaine quantité de la période du motif [21]. La phase de chaque pixel écran étant connue, la mesure de la phase de chaque pixel caméra permet de retracer les rayons incidents et réfléchis, et ainsi faire correspondre les pixels caméra aux pixels écran. La bissectrice de l'angle entre ces deux rayons est la normale à la surface<sup>6</sup>. La phase varie suivant la distance  $d$  entre l'écran et la surface, et suivant la pente locale  $\alpha$  de la surface [20].

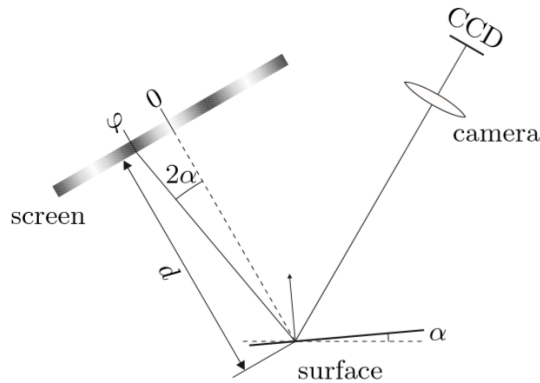
$$\varphi = d \cdot \tan(2\alpha) \quad (1.14)$$

Pour la calibration, il est important de ne pas oublier qu'une pente  $\alpha$  de la surface sous test par rapport à une référence induit un changement de  $2\alpha$  à l'angle de réflexion [20] comme l'illustre la figure 1.19 (conséquence directe de la loi de la réflexion).

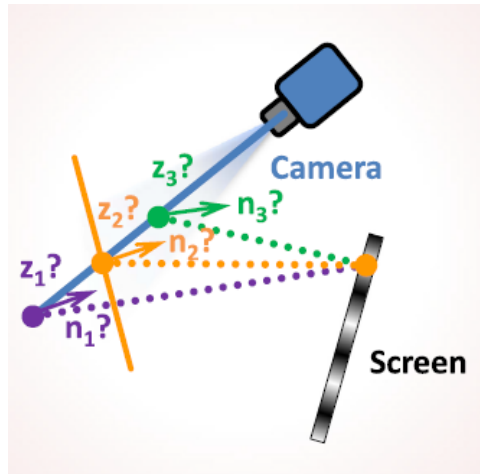
Dans la pratique, le trajet des rayons lumineux est inversé afin de faciliter l'analyse des données. Ainsi, le faisceau sonde partant de la caméra est vu comme la source lumineuse. Cette lumière est réfléchi sur l'écran par l'intermédiaire de la surface. Suivant la formule (1.14), la phase d'un point aperçu par le rayon sonde peut provenir de plusieurs combinaisons hauteur-pente : il s'agit de l'ambiguïté hauteur-pente [31] comme l'illustre la figure 1.20. En effet, la bissectrice de l'angle entre les deux faisceaux doit être la normale d'un point de la surface et non d'un point quelconque de l'espace. Il existe plusieurs moyens pour contrebalancer cette ambiguïté : une estimation préalable de la surface, une utilisation de deux écrans, la stéréo vision ou une approche numérique [31]. Ces trois derniers cas seront considérés comme une amélioration de la PMD et seront expliqués dans la suite du chapitre. De plus, un problème intrinsèque au système limite la technique : la profondeur de champ. En effet, la caméra ne peut pas se mettre au point sur l'objet et sur les franges en même temps. Il y a donc un compromis entre la résolution angulaire et la résolution spatiale durant la capture de l'image. Pour diverses raisons énoncées ci-contre, la caméra sera focalisée sur la surface et non sur les franges car

---

6. La normale se trouve dans le même plan que les faisceaux incidents et réfléchis.



**FIGURE 1.19** – Mesure de la phase  $\varphi$  après réflexion du rayon sur une surface de pente  $\alpha$  par rapport à une référence. [20]



**FIGURE 1.20** – Ambiguïté hauteur-pente. Il existe plusieurs combinaisons haute-pente le long du faisceau sonde qui fournissent la même phase sur l'écran. Cependant, la seule combinaison valable est celle qui donne le point sur la surface. [31]

- La reconstruction 3D d'une image nécessite une bonne résolution spatiale. En effet, la reconstruction de la surface demande de détecter des petits défauts.
- Le calcul de phase est très peu sensible à une légère défocalisation des franges du fait que celles-ci sont des courbes lisses d'intensité. Cette défocalisation provoque une diminution de contraste des franges dont l'incertitude de la mesure sera présentée dans la partie discussion.
- En focalisant sur la surface, les franges deviennent floues. Ce flou permet de lisser les franges sinusoidales qui sont pixelisées.

D'un point de vue expérimental, une mesure PMD s'effectue comme suit :

- La calibration du système. Il s'agit d'un point critique et très important. La caméra, l'écran et la surface doivent se trouver dans une configuration bien particulière et connue très précisément. Il existe plusieurs moyens pour calibrer le dispositif.
- L'acquisition des intensités de franges (verticales et horizontales) réfléchies par la surface à

tester.

- L'analyse des données pour obtenir les valeurs de phase absolue et des pentes en  $x$  et en  $y$ .
- La reconstruction de la surface sur base des données récoltées (processus d'intégration 2D des pentes). Cette étape s'effectue en considérant le trajet inverse de la lumière.

Un résultat important à récupérer pour la construction 3D de la forme de la surface est la phase des franges capturées par la caméra, appelée la phase absolue. Cette information est obtenue en deux grandes étapes<sup>7</sup> [31].

La première s'appelle la démodulation des franges. Cette analyse utilise l'intensité des franges réfléchies pour récupérer les phases enroulées grâce à un algorithme de déphasage. Pour ce faire, l'intensité de chaque pixel de l'écran est modulée par une fonction sinusoïdale. A chaque prise de mesure, celle-ci est décalée d'une fraction de sa période. Ce déphasage s'effectue habituellement en 4 étapes, c'est-à-dire que la fonction sinus est déplacée de  $\frac{\pi}{2}$ . Ainsi, la valeur de la phase d'un pixel est obtenue à l'aide des 4 intensités enregistrées. La formule générale de récupération de la phase enroulée [31] est

$$\varphi^w(x, y) = -\arctan\left[\frac{\sum_{n=0}^{N-1} I_n \sin\left(\frac{2n\pi}{N}\right)}{\sum_{n=0}^{N-1} I_n \cos\left(\frac{2n\pi}{N}\right)}\right] \quad (1.15)$$

Où  $N$  est le nombre d'images,  $\varphi^w(x, y)$  est la phase enroulée aux coordonnées  $(x, y)$  et  $I$  est l'intensité de l'image. Il est à noter que plus le nombre d'étapes dans le déphasage est important, moins le calcul est sensible au bruit. Cependant, les temps d'acquisitions des images deviennent plus longs. Les phases enroulées se trouvent dans l'intervalle  $[-\pi, \pi]$  à cause de l'utilisation la fonction arctangente. Cette limitation s'appelle l'ambiguïté  $2\pi$  [21]. En effet, la fonction sinusoïdale est périodique sur  $2\pi$  et le motif affiché sur l'écran contient plusieurs périodes. La conséquence est que des pixels miroir possèdent la même phase calculée alors qu'il n'en est rien.

Dès lors, la deuxième étape devient indispensable pour trouver les phases réelles des pixels. Elle s'appelle le déroulage de phase. Cette méthode a pour but de prolonger les valeurs de phase enroulée au delà de  $[-\pi, \pi]$ , comme le montre la figure 1.21. Une explication plus approfondie de l'algorithme de déroulage utilisé dans ce mémoire se fera dans le chapitre 3.

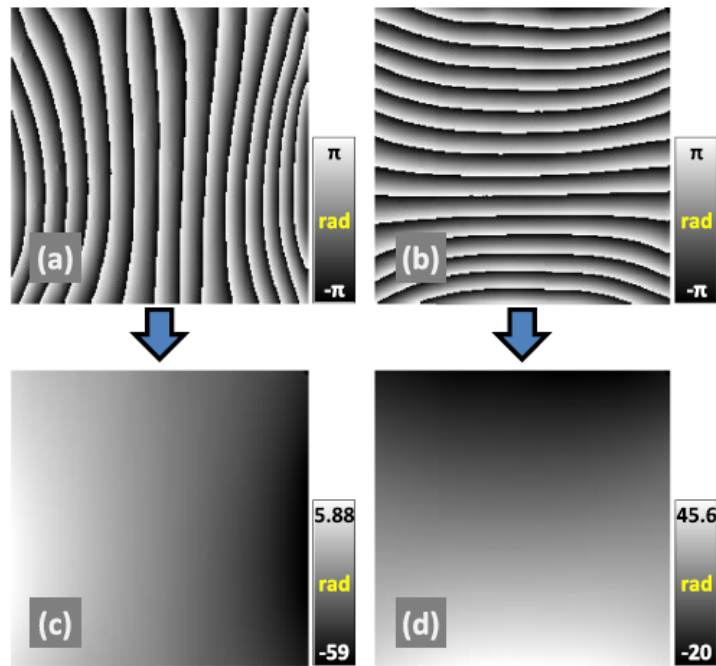
Etant donné que la période du motif affiché sur l'écran et la taille des pixels sont connus, l'emplacement des valeurs de phase absolue sur l'écran devient un paramètre déterminé [21].

$$m = \frac{\phi n}{2\pi p} \quad (1.16)$$

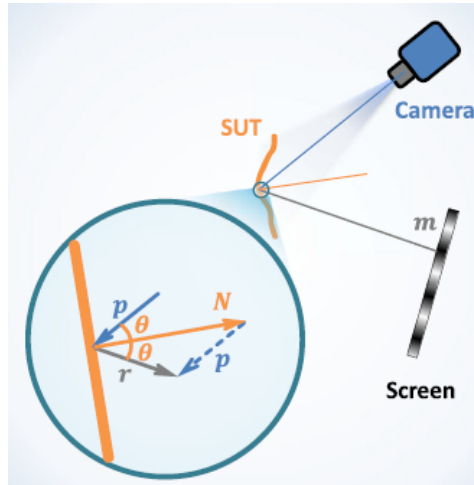
Où  $m = (x_{screen}, y_{screen})$  est la coordonnée du pixel écran,  $p$  est la période du motif et  $n$  est le nombre total de pixels et  $\phi$  la phase déroulée. Les coordonnées des pixels écran sont évaluées dans le système de coordonnées de la caméra. le passage en coordonnées globales s'effectue grâce à la calibration du système. A ce stade, il est possible d'obtenir la normale  $\mathbf{N}$  de la surface comme le montre la figure 1.22.

---

7. Il est à préciser que ce développement doit s'effectuer pour des franges verticales et horizontales. En d'autres mots, la phase absolue est d'abord calculée dans une direction et puis dans l'autre.



**FIGURE 1.21** – Phases enroulées verticales (a) et horizontales (b) et phases déroulées verticales (c) et horizontales (d). Ces phases déroulées portent le nom de phases absolues et sont les phases réelles des pixels.[31]



**FIGURE 1.22** – Obtention de la normale  $\mathbf{N}$  de la surface sous test à partir des vecteurs normalisés sondes  $\mathbf{p}$  et réfléchis  $\mathbf{r}$ . [31]

Le vecteur normal de la surface [31] est

$$\mathbf{N} = \mathbf{r} - \mathbf{p} =: (N_x, N_y, N_z)^T \quad (1.17)$$

Où  $\mathbf{r}$  est le vecteur normalisé réfléchi et obtenu suivant la configuration globale du système et  $\mathbf{p}$  est le vecteur normalisé sonde obtenu au moyen de la calibration de la caméra pour chaque pixel. Les

pentés en x et en y au point  $m$  sont déterminées à l'aide des composantes de la normale [31].

$$s_x = -\frac{N_x}{N_z} \quad (1.18)$$

$$s_y = -\frac{N_y}{N_z} \quad (1.19)$$

De manière générale, la reconstitution de la forme de la surface s'effectue par un processus d'intégration 2D qui fournit la répartition de hauteur  $z$  [31]

$$z = f_{int2}(x, y, s_x, s_y) \quad (1.20)$$

La construction nécessite donc les coordonnées  $(x, y)$  et les pentés calculés  $(s_x, s_y)$ .

#### Discussions :

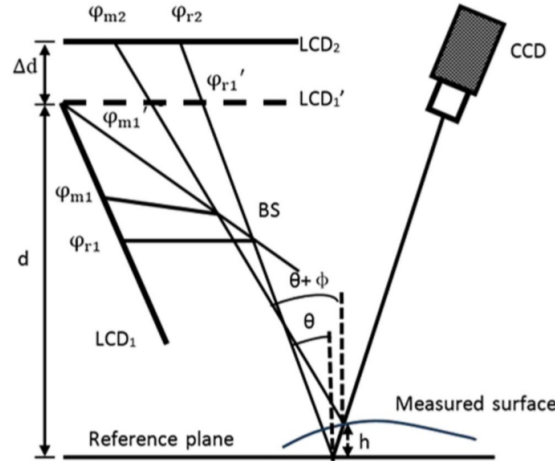
La PMD classique est largement utilisée notamment pour l'algorithme de déphasage en 4 étapes [31]. Cependant, la reconstitution de la surface présente de nombreux problèmes. En effet, la forme absolue de la surface est obtenue en intégrant les pentés calculés [30]. Ces pentés sont obtenues à partir d'une idée préalable de la hauteur de la surface. Ainsi, l'incertitude initiale sur la hauteur entraîne une erreur sur les phases calculés et donc sur les pentés locaux obtenues. De plus, la procédure d'intégration est très sensible aux erreurs et les propage rapidement. De ce fait, des surfaces spéculaires discontinues et/ou isolées de formes complexes ne peuvent être mesurées directement [30]. Contrairement à la méthode de centroïde, la méthode PMD n'est sensible que d'un point de vue temporel aux variations de la lumière [21]. Dès lors, la variation d'intensité entre pixels dans une image n'a pas d'importance. Ce qui impacte les mesures, c'est la façon dont l'intensité varie entre deux images. Cette technique est intéressante pour notre cas d'étude. En effet, l'obtention de cartes de phase permettra de caractériser l'état du système. La problématique sur l'intégration de pentés n'est pas importante, car ce mémoire n'a pas pour but de reconstruire la surface. L'analyse de la phase autrement que pour obtenir les pentés locaux peut être une piste à envisager.

### **Direct Phase Measuring Deflectometry**

Une variante de la PMD est la DPMD (Direct Phase Measuring Deflectometry). Cette méthode a été mise sur pied pour établir une relation directe entre hauteur et phase [30]. Ainsi, la reconstruction 3D de la surface s'effectue sans passer par une technique d'intégration de pentés. Elle permet donc de mesurer des objets spéculaires avec des surfaces isolées et/ou discontinues. Concrètement, quatre paramètres influençant les résultats de mesures ont été identifiés et seront évoqués dans la partie discussion [32].

Pour cette méthode, le dispositif est illustré à la figure 1.23. Le système comprend 2 écrans LCD, notés  $LCD_1$  et  $LCD_2$ , un miroir de référence (pour fournir différents points de vue), une surface à tester, une caméra CCD et un séparateur de faisceau (beamsplitter en anglais et noté BS) [33]. La position du BS doit être convenablement ajustée pour créer une image virtuelle de  $LCD_1$  parallèle au  $LCD_2$ , cette image est noté  $LCD'_1$ . L'utilisation du BS évite de bouger mécaniquement un écran sur deux positions différentes. Le système peut donc être vu comme un écran LCD à deux positions simultanées bien connues. De plus, ces deux écrans doivent aussi être parallèles avec le miroir de référence. Pour contrebalancer le défaut de parallélisme qui subsiste, un motif de franges prédéformées est généré par logiciel [34]. On note  $h$  la hauteur d'un point donné de la surface sous test,  $d$  est la distance entre l'image virtuelle ( $LCD'_1$ ) et le miroir de référence, et  $\Delta d$  est la distance entre les deux écrans  $LCD_2$  et  $LCD'_1$ . Avec un nombre de franges optimales, un motif sinusoïdal identique est

affiché sur les deux écrans. La caméra CCD enregistre, d'un autre point de vue, les motifs déformés suite à la réflexion sur la surface sous test. La phase absolue de chaque pixel écran est obtenue par un algorithme connu appelé l'algorithme de décalage de phase en 4 étapes (évoqué dans la section PMD précédente).



**FIGURE 1.23** – Schéma du dispositif mesurant la surface en utilisant la méthode DPMD. Cette technique est caractérisée par l'utilisation de deux écrans dont l'un (LCD<sub>2</sub>) est parallèle à l'image virtuelle (LCD<sub>1</sub>') de l'autre. Ainsi, l'information sur la hauteur  $h$  de la surface s'obtient directement par la phase calculée. [33]

Si le système est considéré comme un sténopé, le calcul de la hauteur de la surface est réalisé en imaginant deux rayons lumineux incidents [32]. L'un est réfléchi sur le miroir de référence avec un angle  $\theta$  par rapport à la normale et les phases absolues sont notées  $\varphi'_{r1}$  et  $\varphi_{r2}$ . L'autre rayon se réfléchit sur la surface sous test avec un angle  $\theta + \phi$  et les phases absolues sont notées  $\varphi'_{m1}$  et  $\varphi_{m2}$ . Les rayons réfléchis se confondent et sont capturés par la caméra CCD. La hauteur  $h$  de la surface spéculaire par rapport au miroir de référence est obtenue au moyen de relations géométriques et des phases absolues calculées à l'aide des franges réfléchies [33].

$$h = \frac{\Delta d(\varphi_{r1} - \varphi_{m1}) - d[(\varphi_{r1} - \varphi_{r2}) - (\varphi_{m1} - \varphi_{m2})]}{(\varphi_{m1} - \varphi_{m2}) + (\varphi_{r1} - \varphi_{r2})} \quad (1.21)$$

Comme énoncé précédemment, quatre paramètres influencent les mesures :  $d$ ,  $\Delta d$ , la période  $P$  des franges et l'angle d'incidence  $\theta$ .

Les avantages de la DMPD sont multiples. D'une part, la calibration ne fait intervenir aucune pièces en mouvement ce qui implique une meilleure précision durant la prise de mesure. Contrairement à la PMD, la surface est obtenue sans passer par l'intégration d'un gradient de pente ce qui évite toute accumulation d'erreurs. De plus, il devient facile de reconstruire des surfaces spéculaires discontinues et/ou isolées grâce à la détermination de la phase absolue de chaque pixel.

Contrairement à la technique stéréo-PMD expliquée ci-dessous, la DPMD donne l'information sur la profondeur directement à partir des deux cartes de phase absolue venant des deux écrans (dans la stéréo, il n'y a pas deux écrans mais bien deux caméras) [30].

#### Discussions :

Des simulations [32] ont montré que plus la distance  $d$  était petite plus l'erreur quadratique diminue. Ce qui semble intéressant dans ce mémoire étant donné que le système doit être compacte

afin d'être mis en oeuvre sur un CubeSat. Ensuite, l'erreur RMS diminue rapidement avec la distance entre les écrans  $LCD'_1$  et  $LCD_2$ ,  $\Delta d$ . Après une certaine valeur, cette décroissance possède une faible pente car la distance entre les phases voisines sur un même écran augmente. Cette erreur RMS diminue graduellement aussi avec une augmentation de l'angle  $\theta$  entre l'axe optique de la caméra et le plan de référence. Malgré qu'un grand angle  $\theta$  réduit considérablement le bruit, il ne faut toutefois pas que la caméra soit trop inclinée de risque de ne plus capter la lumière réfléchie. L'augmentation de la période  $P$  des franges a pour conséquence une augmentation de l'erreur RMS. De plus, le parallélisme entre les écrans  $LCD'_1$ ,  $LCD_2$  et le plan de référence doit être le plus parfait possible pour s'assurer que les valeurs de  $d$  et  $\delta d$  soient bien celles souhaitées. Chose pas évidente à obtenir et à vérifier lors d'un déploiement de CubeSat dans l'espace. Un autre point à ne pas oublier c'est la quantité de composant. En effet, le but de ce travail est de réaliser un système compact pour un CubeSat. Cependant, cette technique utilise plusieurs éléments : un plan de référence, une caméra, deux écrans et un séparateur de faisceaux. L'alignement de ce dernier est quelque chose de crucial qui ne peut être obtenu facilement dans l'espace.

Pour cette méthode, le rapport signal sur bruit s'améliore avec une augmentation de  $d$ ,  $\Delta d$ ,  $P$  et avec une diminution de  $\theta$ . En effet, le rapport signal sur bruit est amélioré du fait que les distances entre les phases  $\varphi_{r1}$  et  $\varphi_{m1}$ ,  $\varphi_{r1}$  et  $\varphi_{r2}$ ,  $\varphi_{m1}$  et  $\varphi_{m2}$  augmentent.

### Stereo Deflectometry

Il est possible de réaliser la technique de déflectométrie avec deux caméras type CCD [35] : la stéréo déflectométrie dont le principe de mesure est illustré sur la figure 1.24. Les caméras 1 et 2

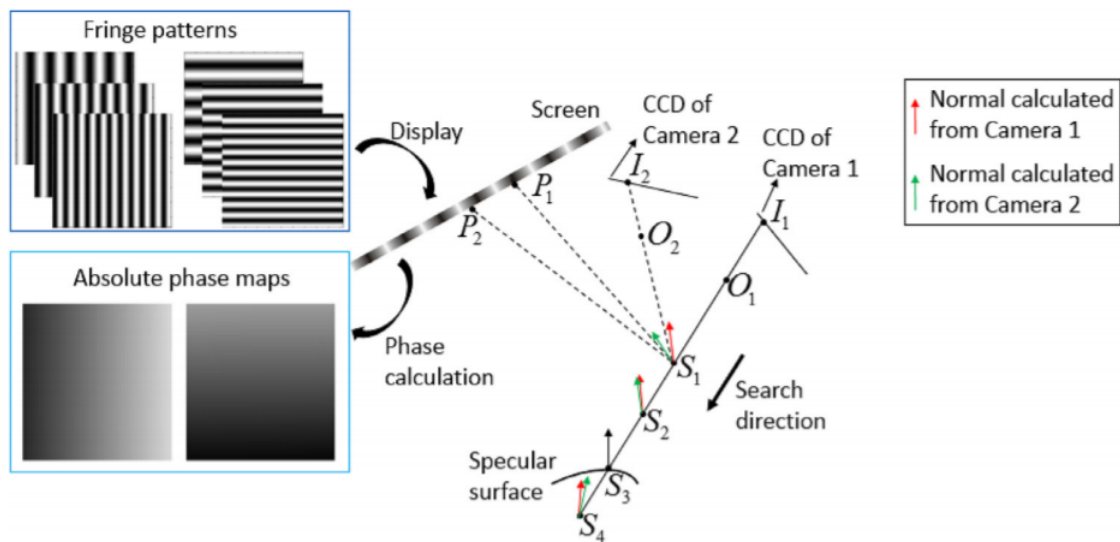


FIGURE 1.24 – Principe de mesure de la stéréo déflectométrie. [35]

possèdent comme centre optique  $O_1$  et  $O_2$  respectivement [35]. Le principe de base est identique aux méthodes précédentes : calcul de phase sur base de franges perpendiculaires. Pour chaque caméra, deux cartes de phase (selon les directions  $x$  et  $y$ ) sont obtenues. Un vecteur normal d'un point  $S_1$  arbitraire de l'espace est obtenu par triangulation avec les points  $S_1, P_1$  et  $I_1$ . Le point  $P_1$  est calculé au moyen des valeurs de phase orthogonales de  $I_1$ . Le point  $I_1$  qui est l'image de  $S_1$  sur la CCD de

la caméra 1 (aussi appelé caméra principale) est connu grâce à la calibration du système. Un autre vecteur normal du point  $S_1$  est obtenu par le même raisonnement avec la caméra 2 (appelé caméra de référence). Une idée première de la surface peut dès lors être effectuée puisqu'en effet les deux vecteurs normaux calculés par les deux caméras se superposent quand le point  $S_1$  est sur la surface de mesure. Comme dans les autres techniques, la formes globale de la surface est reconstruite avec les vecteurs normaux trouvés précédemment. Tous les développements mathématiques concernant le principe et les erreurs ne sont pas montrés dans ce mémoire, mais ils se trouvent dans l'article [35]. L'intérêt ici étant uniquement de se familiariser avec le concept.

#### Discussions :

Les erreurs du système sont influencées par quatre paramètres géométriques [36] :  $\gamma$  qui est l'angle entre la caméra principale et la normale  $\mathbf{n}$  de la surface,  $L_s$  qui est la distance entre l'écran d'affichage et la surface,  $L_d$  qui est la distance entre la caméra principale et la surface, et  $\theta$  qui est l'angle entre les deux caméras. Ces erreurs diminuent en augmentant  $\gamma$ , mais en diminuant  $L_s$ ,  $L_d$  (dans l'ordre du mm) et  $d$ . L'essence même de ce dispositif n'est pas intéressant pour ce mémoire qui, pour rappel, ne cherche pas à reconstruire la surface.

### **Modal Phase Measuring Deflectometry**

La déflectométrie de mesure de phase modale (MPMD) a été introduite pour traiter les données de la PMD sur base de modèles mathématiques prédéfinis [37]. L'idée principale est la reconstruction des pentes et des hauteurs simultanément par l'optimisation des coefficients du modèle pour réduire l'écart entre la mesure effectuée et la reprojction en ray tracing. Cette technique est largement utilisée pour la reconstruction de surface spéculaire de forme libre.

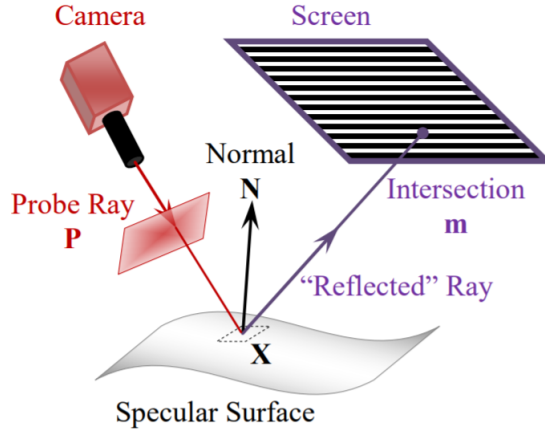
En effet, une surface peut être modélisée par un modèle mathématique, comme par exemple les polynômes de Zernike ou encore les polynômes de Tchebyshev. Le principe de la MPMD est de se donner un de ces polynômes, prenons l'exemple d'un polynôme de Zernike, et ensuite d'effectuer un ray tracing avec la surface décrite par ces polynômes. On compare alors les franges obtenues numériquement avec les franges mesurées dans l'expérience. La correspondance entre les deux images s'effectue en modifiant les coefficients des polynômes. Ainsi, une fois que les franges simulées et mesurées coïncident, la topologie de la surface est directement connue grâce aux coefficients. En pratique, un nombre de termes finis des polynômes est utilisé entachant la comparaison d'une imprécision. Dès lors, la surface réelle n'est jamais parfaitement décrite par le modèle mathématique. La MDPD fournit néanmoins une méthode pour minimiser ces erreurs.

Le développement ci-dessous explique la marche à suivre ainsi que les grandes étapes de calculs pour optimiser au mieux les coefficients de Zernike et montrer comment minimiser les erreurs dues au nombre de termes choisi dans les polynômes. L'article [37] propose une démonstration plus en détails que ce qui est exposé ci-dessous et les formules sont toutes tirées de cet article.

Comme dans toutes les autres techniques, les trajets des rayons lumineux sont traités comme provenant de la caméra. La figure 1.25 illustre la caméra, le rayon sonde issu de celle-ci  $\mathbf{P} = (x_n, y_n, 1)^T$ ,  $\mathbf{X} = (x_c, y_c, z_c)^T = z_c \mathbf{P}$  le point d'intersection entre le rayon sonde et la surface, et  $\mathbf{m}$  le point d'intersection entre le rayon réfléchi et l'écran.

Le vecteur  $\mathbf{w}$  est ici un vecteur où chacune de ses composantes est une fonction de base des polynômes choisis. La longueur de ce vecteur est égale au nombre de termes choisis. Ainsi, si la surface est modélisée par les polynômes de Zernike, chaque composante de  $\mathbf{w}$  correspond à une fonction de Zernike. Les polynômes de Zernike s'expriment donc comme  $\mathbf{w}^T \mathbf{c}$  où  $\mathbf{c}$  est le vecteur des coefficients. La convergence entre les deux surfaces se fait donc en modifiant ces coefficients  $\mathbf{c}$ .

Dans les coordonnées normalisées de la caméra, on a



**FIGURE 1.25** – Dispositif de déflectométrie à mesure de phase en considérant les rayons comme issus de la caméra. Ceux-ci portent le nom de rayon sonde. [37]

$$z_c = \mathbf{w}^T \mathbf{c} \quad (1.22)$$

$$\frac{\partial z_c}{\partial x_n} = \mathbf{w}_x^T \mathbf{c} \quad (1.23)$$

$$\frac{\partial z_c}{\partial y_n} = \mathbf{w}_y^T \mathbf{c} \quad (1.24)$$

Où  $\mathbf{w}_x^T$  et  $\mathbf{w}_y^T$  sont les dérivées des fonctions de base correspondantes du modèle. Comme montré dans [37], en développant ces dérivées partielles  $\frac{\partial z_c}{\partial x_n}$  et  $\frac{\partial z_c}{\partial y_n}$ , les pentes de la surface mesurées peuvent s'exprimer comme

$$\frac{\partial z_c}{\partial x_c} = \frac{\frac{\partial z_c}{\partial x_n}}{z_c + x_n \frac{\partial z_c}{\partial x_n} + y_n \frac{\partial z_c}{\partial y_n}} \quad (1.25)$$

$$\frac{\partial z_c}{\partial y_c} = \frac{\frac{\partial z_c}{\partial y_n}}{z_c + x_n \frac{\partial z_c}{\partial x_n} + y_n \frac{\partial z_c}{\partial y_n}} \quad (1.26)$$

Ces pentes en  $x$  et en  $y$  permettent de décrire la normale  $\mathbf{N} = \left( \frac{\partial z_c}{\partial x_c}, \frac{\partial z_c}{\partial y_c}, -1 \right)$ . Une fois la normale construite pour un point  $\mathbf{X}$  et un rayon sonde  $\mathbf{P}$ , le rayon réfléchi  $\mathbf{r}$  se construit à l'aide des lois de la réflexion

$$\mathbf{r} = \mathbf{p} - 2\langle \mathbf{p}, \mathbf{n} \rangle \mathbf{n} \quad (1.27)$$

Avec  $\mathbf{n} = \frac{\mathbf{N}}{\|\mathbf{N}\|}$  et  $\mathbf{p} = \frac{\mathbf{P}}{\|\mathbf{P}\|}$ . L'intersection entre le rayon réfléchi et le miroir est obtenue en connaissant la position relative du miroir comparé à la caméra. Cette position relative peut être décrite par un vecteur rotation  $\boldsymbol{\omega}$  et un vecteur de translation  $\mathbf{T}$ . Ceux-ci peuvent être connus grâce à une phase de pré-calibration classique comprenant une calibration de l'écran, de la caméra et de la géométrie. L'étape suivante est le calcul du point  $\hat{\mathbf{m}}$  qui est l'intersection entre le rayon réfléchi obtenu par ray tracing et le miroir, et le comparer avec le point réel  $\mathbf{m}$  obtenu pour le même rayon sonde. Ainsi, les coefficients  $\mathbf{c}$  optimaux (ainsi que la position exacte de l'écran, c'est-à-dire  $\boldsymbol{\omega}$  et  $\mathbf{T}$ )

peuvent être obtenus en minimisant l'écart entre  $\mathbf{m}$  et  $\hat{\mathbf{m}}$ . On peut effectuer cette minimisation en résolvant le problème de moindre carré suivant :

$$[\hat{\omega}, \hat{\mathbf{T}}, \hat{\mathbf{c}}] = \operatorname{argmin}_{\omega, \mathbf{T}, \mathbf{c}} \sum_{n=1}^N \|\hat{\mathbf{m}}_n(\omega, \mathbf{T}, \mathbf{c}) - \mathbf{m}_n\|^2 \quad (1.28)$$

où  $N$  est le nombre de pair valide. En utilisant un algorithme comme l'algorithme de Levenberg-Marquardt, il est possible de déterminer les paramètres par itérations obtenant ainsi la meilleure forme des polynômes de Zernike pour décrire cette surface. Néanmoins, comme mentionné plus haut, au vu du nombre fini de terme, il subsistera une erreur entre la surface réelle et celle décrite par le modèle mathématique. Cette erreur peut néanmoins être caractérisée et être réduite. Le résidu sur la position de l'intersection entre le rayon réfléchi et le miroir est  $\Delta \mathbf{m} = \mathbf{m} - \hat{\mathbf{m}}$ . A partir de  $\mathbf{m}$  et  $\hat{\mathbf{m}}$ , il est également possible de calculer cet écart entre le modèle et la réalité pour la normale calculée par ray tracing  $\hat{\mathbf{N}}$  et la vraie normale  $\mathbf{N}$

$$\hat{\mathbf{N}} = \frac{\hat{\mathbf{m}} - \hat{\mathbf{X}}}{\|\hat{\mathbf{m}} - \hat{\mathbf{X}}\|} - \frac{\mathbf{P}}{\|\mathbf{P}\|} \quad (1.29)$$

$$\mathbf{N} = \frac{\mathbf{m} - \hat{\mathbf{X}}}{\|\mathbf{m} - \hat{\mathbf{X}}\|} - \frac{\mathbf{P}}{\|\mathbf{P}\|} \quad (1.30)$$

Il en va de même pour les pentes aux points  $\hat{\mathbf{X}}$  et les pentes résiduelles  $(\Delta S_x, \Delta S_y)$

$$\Delta S_x = \tan(\arctan(\hat{S}_x) - \theta_y) - \hat{S}_x \quad (1.31)$$

$$\Delta S_y = \tan(\arctan(\hat{S}_y) - \theta_x) - \hat{S}_y \quad (1.32)$$

où  $\hat{S}_x$  et  $\hat{S}_y$  sont les pentes estimées par le modèle, et  $\theta_x$  et  $\theta_y$  sont les angles de rotation d'une normale par rapport à l'autre, ainsi ces angles peuvent être calculés à partir des normales  $\hat{\mathbf{N}}, \mathbf{N}$  [37]. Des méthodes d'intégrations 2D, comme par exemple les méthodes d'intégrations zonales peuvent alors être utilisées pour reconstruire la hauteur à partir des pentes résiduelles,

$$\Delta z = f_{\text{zonal}}(\hat{x}, \hat{y}, \Delta S_x, \Delta S_y) \quad (1.33)$$

La hauteur  $z$  de la surface en un point est la hauteur estimée par les polynômes de Zernike  $\hat{z}$  plus la correction  $\Delta z$  :

$$z = \hat{z} + \Delta z \quad (1.34)$$

Ainsi par la méthode MPMD nous avons une bonne estimation de la surface.

#### Discussions :

L'approche de cette technique pour traiter la problématique est assez intéressante dans le cas de ce mémoire. En effet, la MPMD obtient les informations souhaitées en se basant sur les coefficients des polynômes. Toute la caractérisation du système repose sur les valeurs finales des coefficients. Pour ce mémoire, ces polynômes peuvent utiliser la signature des coefficients de polynômes pour caractériser son désalignement dans l'espace. En combinant cette idée avec la PMD, il serait possible d'imaginer de tirer une signature en coefficients des cartes de phase absolue.

## Chapitre 2

# Mise en oeuvre au laboratoire

Ce chapitre traite des erreurs liées à l'expérimentation, mais également à toute la mise en oeuvre d'un dispositif de déflectométrie. Il est important de spécifier qu'au vu de la crise pandémique COVID-19 que traverse pour le moment notre société, les deux miroirs n'ont pas été livrés en même temps. Ce chapitre se basera donc sur des mesures réalisées avec un seul miroir. De plus, les expériences n'ont pas pu être menées à bien étant donné la fermeture du laboratoire au cours de l'année. L'idée de ce chapitre expérimental est donc de montrer comment corriger une des erreurs évoquées dans le cas de l'expérimentation.

### 2.1 Erreurs systématiques

Les erreurs systématiques sont des erreurs liées à l'utilisation de matériel pour l'expérience. Leurs effets sur les valeurs sont totalement indépendants du nombre de mesures effectuées. Leurs connaissances sont importantes afin de calibrer au mieux un dispositif expérimental.

#### 2.1.1 Erreur sur la phase

Une première source d'erreur est **la quantification du signal** [21]. En effet, une fonction continue, comme la fonction sinus, est numérisée de façon discrète à cause du nombre de bits utilisés. Il est possible d'évaluer cette erreur en simulant une fonction sinus non échantillonnée et de la comparer avec un échantillonnage. Le résultat est illustré à la figure 2.1.<sup>1</sup> La courbe en rouge est le sinus non échantillonné, les courbes sinus en 8, 7 et 6 bits sont respectivement en bleu, vert et rose. Le codage sous 8 bits semble être le mieux adapté pour minimiser l'erreur de quantification. Cette erreur va donc impacter la précision sur la valeur de la phase car de l'information est perdue. Ce choix de 8 bits et non de bits plus élevés se justifie par un soucis de gain de temps. Les fichiers seront plus légers et le traitement des images plus rapides.

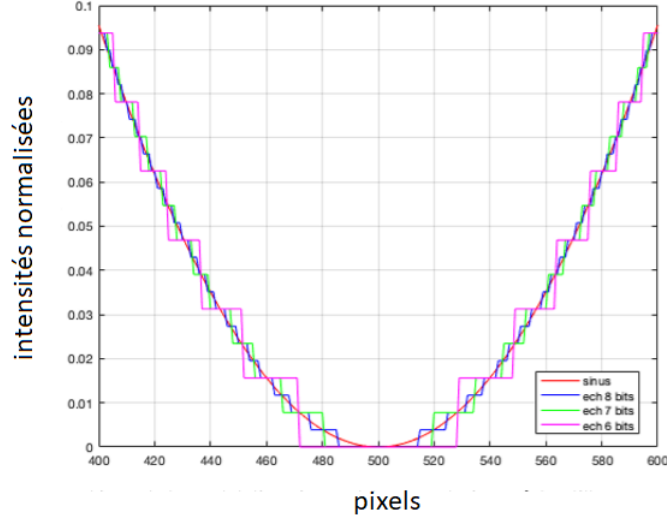
De plus, l'erreur engendrée va également dépendre du nombre d'étapes réalisées dans le déphasage du sinus. L'écart type de l'erreur [21] est

$$\sigma_q = \frac{2}{\sqrt{3n2^b}} \quad (2.1)$$

Où  $\sigma_q$  est l'écart type de l'erreur de quantification exprimé en radian,  $b$  est le nombre de bits utilisé pour numériser les intensités et  $n$  est le nombre d'étapes du déphasage.

---

1. Ce graphique a été réalisé par mon prédécesseur Alexis Bechet.



**FIGURE 2.1** – Fonction sinus (en rouge) codée en 8 (en bleu), 7 (en vert) et 6 (en rose) bits. Le codage sous 8 bits se rapproche au mieux de la véritable fonction.

Une autre erreur est l'**incertitude des algorithmes de déphasage**. Pour mieux comprendre ce point, il faut revenir sur l'état de la surface. En effet, une surface n'est pas parfaitement spéculaire. D'après son degré de polissage, il subsiste plus ou moins des rugosités la rendant également diffuse. Ainsi, une fraction de la lumière est réfléchie de manière diffuse et l'autre de manière spéculaire vers la caméra [38]. Pour la déflectométrie, seule la partie spéculaire est importante car elle contient l'information sur les franges. La réflexion diffuse a donc comme conséquence de diminuer le contraste des franges, réduisant ainsi la précision dans les mesures. Le motif généré sur l'ordinateur est [31]

$$I_k(x, y) = A(x, y) + B(x, y) \cdot \cos\left(\phi + \frac{2k\pi}{2}\right) \quad (2.2)$$

où  $I_k(x, y)$  est l'intensité du pixel  $(x, y)$ ,  $k$  est le numéro de l'image, c'est-à-dire l'étape de déphasage,  $A(x, y)$  est l'intensité de fond et  $B(x, y)$  est l'amplitude du cosinus.

Ce motif sera capturé par la caméra sous la forme [38]

$$I'_k(x, y) = f[\alpha_s I_k(x, y) + \alpha'_d I_k(x, y)] + E_k(x, y) \quad (2.3)$$

où  $I'_k(x, y)$  est l'intensité bruitée capturée par la caméra,  $f[x]$  est une distribution non linéaire représentant la réponse du système,  $\alpha_s$  est la partie de la lumière réfléchie de manière spéculaire,  $\alpha'_d$  est une portion de la réflexion diffuse  $\alpha_d$  et  $E_k(x, y)$  est le bruit de la caméra. Il est à noter que les rayons diffus atteignant le pixel  $(x, y)$  de la caméra proviennent de réflexions diffuses sur toute la surface tandis que le rayon spéculaire est issu d'une réflexion en un point du miroir. Si les effets non linéaires du système ne sont pas présents (cf. 2.4.3 calibration écran pour la correction de ces effets), l'équation (2.3), en développant les termes, s'exprime comme [38]

$$I'_k(x, y) = \eta\alpha_s A(x, y) + \eta\alpha_d A_d(x, y) + E_k(x, y) + \eta\alpha_s B(x, y) \cos\left(\phi + \frac{2k\pi}{2}\right) \quad (2.4)$$

Où les trois premiers termes sont considérés comme la nouvelle intensité de fond,  $A_d(x, y) = \alpha'_d I_k(x, y)$  et  $\eta$  est une constante définie par [38] comme étant égale à (contraste des franges capturées par la caméra)/(contraste des franges générées sur l'écran).

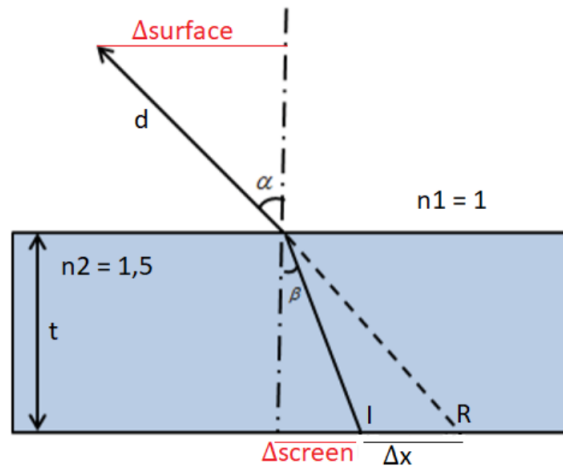
L'extraction de la phase à l'aide d'un algorithme de déphasage à 4 étapes ne fournit qu'une phase bruitée  $\phi'$ . Cependant, celle-ci peut être reliée à la phase sans bruit  $\phi$  par [38]

$$\tan(\phi') = \frac{I'_1 - I'_3}{I'_0 - I'_2} = \frac{\sin(\phi) + \frac{E_1 - E_3}{\eta\alpha_s B(x,y)}}{\cos(\phi) + \frac{E_0 - E_2}{\eta\alpha_s B(x,y)}} \quad (2.5)$$

Une erreur aléatoire sur la phase est donc commise. Elle est tout simplement donnée par la différence entre les deux types de phase. Cette erreur sur la phase est influencée par 4 facteurs [38] :  $E_k(x, y)$ ,  $\eta$ ,  $\alpha_s$  et  $B(x, y)$ . Cette erreur aléatoire diminue en augmentant  $\alpha_s$ , c'est-à-dire augmenter la réflexion spéculaire d'une surface. Elle diminue également en augmentant  $B(x, y)$ . Pour ce faire, la luminosité de l'écran doit être grande et le contraste important, c'est-à-dire maximiser la différence entre une frange sombre (niveau de gris 0) et une frange claire (niveau de gris 255). Il est également possible d'augmenter  $\eta$  en diminuant la défocalisation des franges.

### 2.1.2 Erreurs liées à l'écran

L'utilisation d'un écran n'est pas sans conséquence. En effet, les cristaux liquides d'un écran sont entourés de plusieurs couches de matériaux protecteurs transparents [39]. L'image enregistrée par la caméra ne se situe donc pas à la surface d'un écran, mais bien sous une certaine épaisseur par rapport à la surface. Cette couche de l'écran entraîne **une réfraction** des rayons lumineux suite à l'existence du dioptre air-couche protectrice. Dès lors, la connaissance de la position exacte de chaque pixel écran s'en retrouve erronée. La figure 2.2 illustre bien cette problématique de réfraction. La caméra voit le pixel I alors que sans la couche d'épaisseur  $t$ , elle devrait observer le pixel R. Le décalage  $\Delta x$  entre ces deux pixels est donné par [39]. Pour mieux comprendre l'expression, nous allons la redémontrer.



**FIGURE 2.2** – Déplacement du motif lumineux dans un écran suivant la loi de Snell appliquée pour chaque pixel. Les différentes couches de matériaux transparents sont considérées comme une seule couche d'épaisseur  $t$  afin de faciliter l'analyse de la réfraction. L'indice de réfraction de ce milieu va être assimilé à celui du verre, c'est-à-dire égale à 1,5. Ce phénomène entraîne une déviation du faisceau. La caméra qui est supposée voir le pixel R perçoit réellement le point I. La perception de la position du pixel est décalée de  $\Delta x$ . [39]

La démonstration fait intervenir indéniablement la loi de Snell.

$$n_1 \sin(\alpha) = n_2 \sin(\beta) \quad (2.6)$$

Où  $n_1$  et  $n_2$  sont respectivement les indices de réfraction du milieu 1 et 2,  $\alpha$  (aussi appelé angle d'illumination) et  $\beta$  sont respectivement les angles de déviation par rapport à la normale du milieu 1 et 2. Dans la situation étudiée, le milieu 1 est l'air, ce qui signifie que son indice de réfraction vaut 1. Pour trouver une expression du décalage  $\Delta x$  du pixel, les relations trigonométriques seront utilisées. La démonstration part de l'étude de la distance entre la normale et le rayon  $d$  à une hauteur  $t$

$$\begin{aligned} \Delta_{surface} &= t \tan(\alpha) \\ \Delta_{screen} &= t \tan(\beta) \end{aligned} \quad (2.7)$$

Ainsi, il est aisé de dire

$$\begin{aligned} \Delta x &= \Delta_{surface} - \Delta_{screen} \\ &= t(\tan(\alpha) - \tan(\beta)) \\ &= t(\tan(\alpha) - \tan(\sin^{-1}(\sin(\beta)))) \\ &= t(\tan(\alpha) - \tan[\sin^{-1}(\frac{\sin(\alpha)}{n_2})]) \end{aligned} \quad (2.8)$$

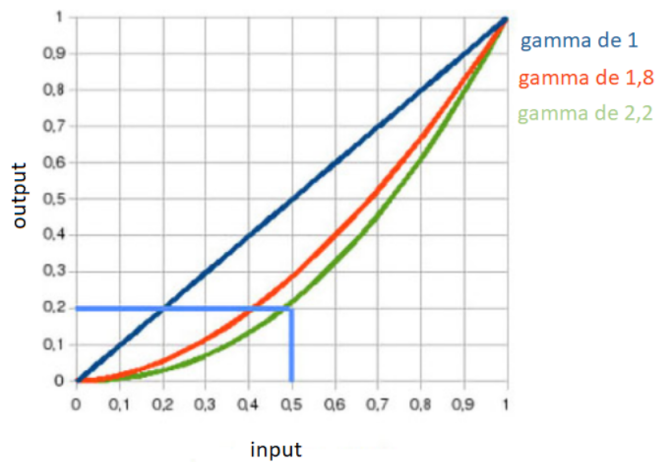
Le passage à la dernière équation s'effectue en remplaçant  $\sin(\beta)$  par l'expression (2.6).  $\Delta x$  est le décalage,  $t$  est l'épaisseur de la couche protectrice,  $n_2$  est l'indice de réfraction et  $\alpha$  est l'angle d'illumination. L'incertitude quant à la position des pixels augmente donc avec  $n$  et  $t$ .

Il est possible de pallier à ce problème soit en étalonnant le système avec une surface de référence, soit en mesurant l'épaisseur  $t$  de la couche à l'aide d'une sonde optique [39].

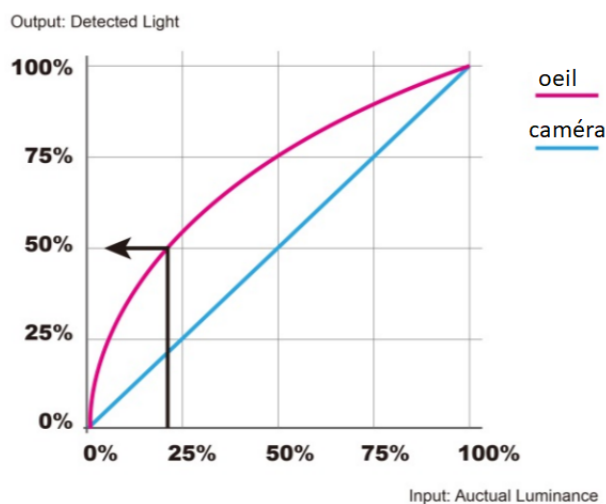
Une deuxième erreur systématique est **la forme** de l'écran. En effet, celui-ci peut ne pas être parfaitement plat suite à sa manufacture.

Une dernière erreur liée à l'écran est sa **non-linéarité** [39]. Ce phénomène se traduit par la courbe mathématique de la figure 2.3, appelée courbe gamma. L'expression de cette courbe est une simple loi exponentielle caractérisée par la valeur gamma :  $output = input^\gamma$  [40].

L'écran n'affiche pas la même luminosité que celle qu'il reçoit en entrée. Il va afficher des images plus contrastées et plus sombres que les images d'entrées. Cette non-linéarité de l'affichage a pour but de coller au mieux à la perception de l'oeil humain [40]. En effet, l'oeil humain possède également un gamma. De plus, la non-linéarité de l'oeil dépend de l'environnement. Lorsque l'oeil regarde un écran d'ordinateur, son gamma est de 1/2, 2, par contre pour la lecture d'un livre il est de 1/1, 8. Cependant, la caméra ne réagit pas comme l'oeil [39] ainsi que l'illustre la figure 2.4. Les images affichées sur l'écran ne sont donc pas adaptées pour un détecteur. Dans ce cas, il est crucial de réaliser une correction gamma. Le but est d'inverser la courbe gamma de l'écran (cette courbe inverse porte le nom de courbe de transfert) pour calibrer la système et ainsi obtenir une relation linéaire entre l'écran et la caméra.



**FIGURE 2.3** – Courbe gamma d’un écran. Le  $\gamma$  standard d’un Mac est de 1,8 et de Windows est de 2,2 [41]. La valeur idéale de  $\gamma$  pour avoir un transfert linéaire est de 1 ainsi les valeurs d’entrées et de sorties seront identiques. [40]



**FIGURE 2.4** – Courbe gamma d’un oeil humain et d’une caméra. La caméra ne réagit pas de la même façon que l’oeil face à la luminosité. Il est à noter que la courbe de l’oeil est l’inverse de la courbe gamma d’un écran. [41]

### 2.1.3 Erreurs liées à la caméra

Un problème inhérent avec l’utilisation d’une caméra est **la distorsion**. Il existe trois types de distorsions : radiale, de centrage et de prisme [39]. Ce phénomène a pour conséquence de décaler les coordonnées de l’objet observé. Cet effet traduit un problème du grandissement. De manière générale, la distorsion provoquée par les lentilles s’exprime [42]

$$\begin{aligned} u_d &= u + \delta_u(u, v) \\ v_d &= v + \delta_v(u, v) \end{aligned} \tag{2.9}$$

Où  $u_d$  et  $v_d$  sont les coordonnées distordues,  $u$  et  $v$  les coordonnées non distordues et  $\delta_u(u, v)$  et  $\delta_v(u, v)$  sont les décalages provoqués par la distorsion. Ainsi, une ligne droite, suivant sa position dans l'objet, peut être vue comme une courbe dans l'image capturée par la caméra.

Le premier type de distorsion est la distorsion radiale. Son origine est une courbure non parfaite de la lentille [43]. Son effet est illustré à la figure 2.5.

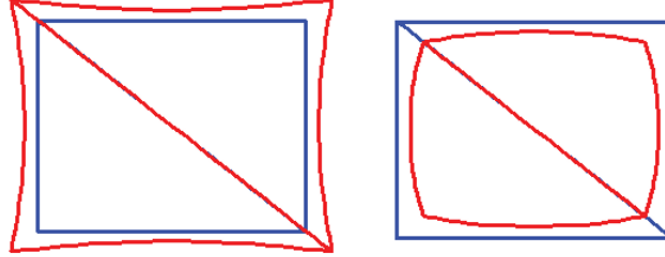


FIGURE 2.5 – Distorsion radiale positive (à gauche) et négative (à droite) . [39]

Celle-ci est gouvernée par l'équation [42]

$$\begin{aligned}\delta_{ur}(u, v) &= u(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \\ \delta_{vr}(u, v) &= v(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots)\end{aligned}\quad (2.10)$$

Où  $r$  est la distance radiale du point  $(u, v)$  par rapport au centre de distorsion (généralement le centre de l'image).  $k_1$ ,  $k_2$  et  $k_3$  sont les coefficients de distorsion radiale.

Le deuxième type de distorsion est le décentrage qui, comme son nom l'indique, provient du décentrage des lentilles dans la caméra. Elles sont dites non colinéaires [43] et provoquent des distorsions tangentielles et radiales. L'effet sur l'image est illustré à la figure 2.6. Son expression est [42]

$$\begin{aligned}\delta_{ud}(u, v) &= p_1(3u^2 + v^2) + 2p_2.u.v = p_1(u^2 + v^2) + u(2p_1.u + 2p_2.v) \\ \delta_{vd}(u, v) &= p_2(3v^2 + u^2) + 2p_2.u.v = p_2(v^2 + u^2) + v(2p_1.u + 2p_2.v)\end{aligned}\quad (2.11)$$

Où  $p_1$  et  $p_2$  sont les coefficients du décentrage.

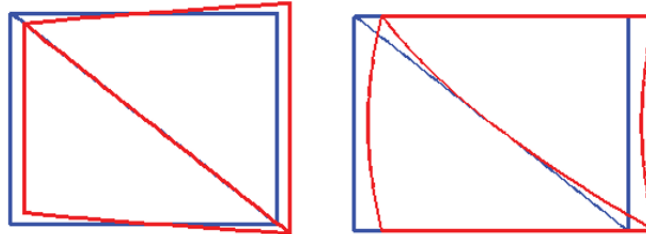


FIGURE 2.6 – Distorsion de décentrage lorsque  $p_2$  est nul (à gauche) et lorsque  $p_1$  est nul (à droite). [39]

Le dernière distorsion est la distorsion de prisme. Son effet n'est pas des plus importants dans l'image et est provoqué suite à un mauvais alignement des composants de la caméra. L'expression

mathématique de cette distorsion est [42]

$$\begin{aligned}\delta_{up}(u, v) &= s_1(u^2 + v^2) \\ \delta_{vp}(u, v) &= s_2(v^2 + u^2)\end{aligned}\tag{2.12}$$

Avec  $s_1$  et  $s_2$  les coefficients de distorsion de prisme.

Dès lors, la distorsion totale d'une image suite à sa capture par la caméra peut être quantifiée de la sorte [42].

$$\begin{aligned}\delta_u(u, v) &= \delta_{ur}(u, v) + \delta_{ud}(u, v) + \delta_{up}(u, v) \\ \delta_v(u, v) &= \delta_{vr}(u, v) + \delta_{vd}(u, v) + \delta_{vp}(u, v)\end{aligned}\tag{2.13}$$

Pour connaître les paramètres internes de la caméra utilisée, la communauté de Matlab a mis au point une boîte à outils, appelé Camera Calibration Toolbox [44]. Elle consiste à analyser des images d'un damier prises sous différents angles et distances par rapport à la caméra. Avec la connaissance de la taille de chaque damier, les algorithmes proposés dans la Toolbox ressortent les paramètres de la caméra.

Une autre problème est **la saturation** [39]. En effet, il se peut que le motif d'entrée soit d'une intensité trop importante pour le détecteur qui finit par saturer à une valeur sans pouvoir détecter des intensités plus élevées. Ce phénomène ne peut être pallier sans perte d'informations. Il y a deux possibilités pour atténuer ou régler cet effet. La première est de diminuer le gain du détecteur. La deuxième consiste à ne travailler que dans la gamme d'intensité ne produisant pas cet effet.

## 2.2 Matériel

Pour réaliser une expérience d'alignement, il faut tout d'abord des miroirs. Le choix des miroirs paraboliques hors axe (noté OAPM pour Off-Axis Parabolic Mirror en anglais) s'est basé sur des caractéristiques réelles de CubeSats. La figure 2.7 reprend les caractéristiques des miroirs choisis dont certains points vont être précisés [45].

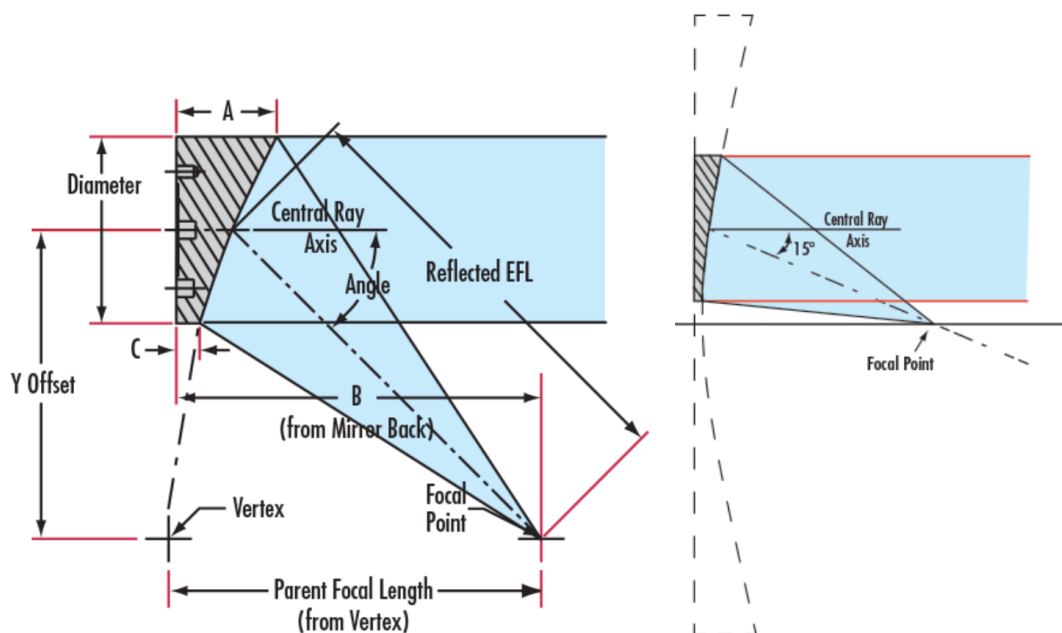
Type:	Off-Axis Parabolic Mirror
Coating Specification:	R <sub>avg</sub> >85% @ 400 - 700nm R <sub>avg</sub> >90% @ 400 - 2000nm
Effective Focal Length EFL (mm):	387.60
Diameter (mm):	50.80 +0.00/-0.38
Parent Focal Length PFL (mm):	381.00
Surface Roughness (Angstroms):	<100 RMS
Clear Aperture (%):	90
Substrate:	Aluminum 6061-T6
Reflected Wavefront, RMS:	1λ
Coating:	Protected Aluminum (400-2000nm)
Coating Type:	Metal
Focal Length Tolerance (%):	±1
Off-Set Angle (°):	15
Surface Figure, RMS:	λ/2
Wavelength Range (nm):	400 - 2000
Surface Quality:	80-50
Y Offset (mm):	100.32

**FIGURE 2.7** – Tableau reprenant les caractéristiques des miroirs paraboliques hors axe choisis. Leur connaissance est cruciale pour mettre en place le système. [45]

Il s'agit de miroirs dont l'angle off-set est de 15 °. Autrement dit, ils représentent des morceaux proches du vertex de la parabole parente<sup>2</sup> de telle façon qu'une lumière venant de l'infini sera déviée de 15 ° vers le point focal, comme l'illustre la figure 2.8.

---

2. La parabole parente est le nom donné à la parabole dont est issu la parabole hors axe.



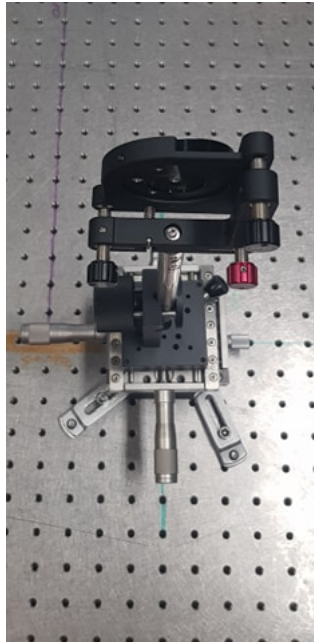
**FIGURE 2.8** – A gauche : Géométrie d'un miroir parabolique hors axe avec les différents termes techniques. A droite : principe d'un miroir parabolique hors axe de  $15^\circ$ . La lumière venant de l'infini sera déviée de  $15^\circ$  par le miroir vers le point focal de la parabole parente. [46]

La distance focale parente (PFL) est la distance focale de la parabole dont est issu l'OAPM. Celle-ci est de 381 mm pour les miroirs sélectionnés. Par contre, la distance focale effective (EFL) est la distance focale des OAPM qui vaut 387,60 mm. La dénomination y offset fait référence à la distance entre l'axe central de l'OAPM et l'axe de la parabole parente. Dans notre cas, cette distance est de 100,32 mm. Toutes ces données sont cruciales pour placer les différents éléments du dispositif. La gamme de longueur d'onde, pour que ces OAPM soient efficaces, se trouve dans le visible. L'expérience sera basée sur l'affichage de franges dans différents niveaux de gris.

Ces OAPM doivent être soutenus à l'aide de montures cinétiques à 3 vis de même diamètre que les miroirs [47]. Les vis vont permettre de placer au mieux les miroirs en appliquant des rotations de la surface autour de ces axes. Afin de réaliser les autres mouvements dans l'espace, des plaques de translations sont utilisées. Le montage est illustré à la figure 2.9.

L'écran choisi est un moniteur LCD Philips avec un écran anti-reflet afin d'éviter que le miroir ne réfléchisse son propre reflet à la caméra. Cet écran mesure 34 cm de long et 27 cm de large avec une résolution de 1280 x 1024 pixels.

La caméra DCC3240C utilisée dans ce montage est une caméra CMOS couleur et monochrome Thorlabs de résolution de 1280 x 1024 pixels. Pour plus d'informations sur la caméra cf. [48].



**FIGURE 2.9** – Miroir parabolique hors axe sur sa monture cinétique avec les trois vis de rotation placées derrière celui-ci. Le tout est monté sur trois plaques de translation qui fournissent un mouvement dans les trois degrés de liberté.

## 2.3 Configuration

La configuration choisie est illustrée sur la figure 2.10. L'écran et la caméra seront placés au centre de courbure de la parabole parente, soit à 762 mm du vertex. Les OAPM sont séparés de 200 mm avec à mi-chemin le vertex. Comme mentionné au début du chapitre, le montage est réalisé avec un seul miroir. Cependant, l'emplacement du second miroir sera pris en compte lors de l'élaboration du dispositif sur la table optique.

Cette configuration a pour conséquence de créer une double imagerie dans le système. La figure 2.11 illustre ce qui se passe au niveau de la caméra avec les rayons lumineux. La première imagerie est fournie par les rayons mauves. Il s'agit de la situation où la caméra se focalise sur le miroir. Ainsi, les rayons issus de l'écran viennent frapper la surface et sont réfléchis vers la caméra. Dans ce cas, la lentille objectif intervient dans la propagation des rayons pour converger les rayons et faire apparaître le miroir net. La seconde imagerie est donnée par les rayons verts. Les franges sur le miroir constituent des points sources. Les rayons issus des franges sur le miroir convergent vers le détecteur sans l'intervention de la lentille objectif. En effet, cette imagerie a lieu uniquement parce que la caméra se situe au centre de courbure du système.

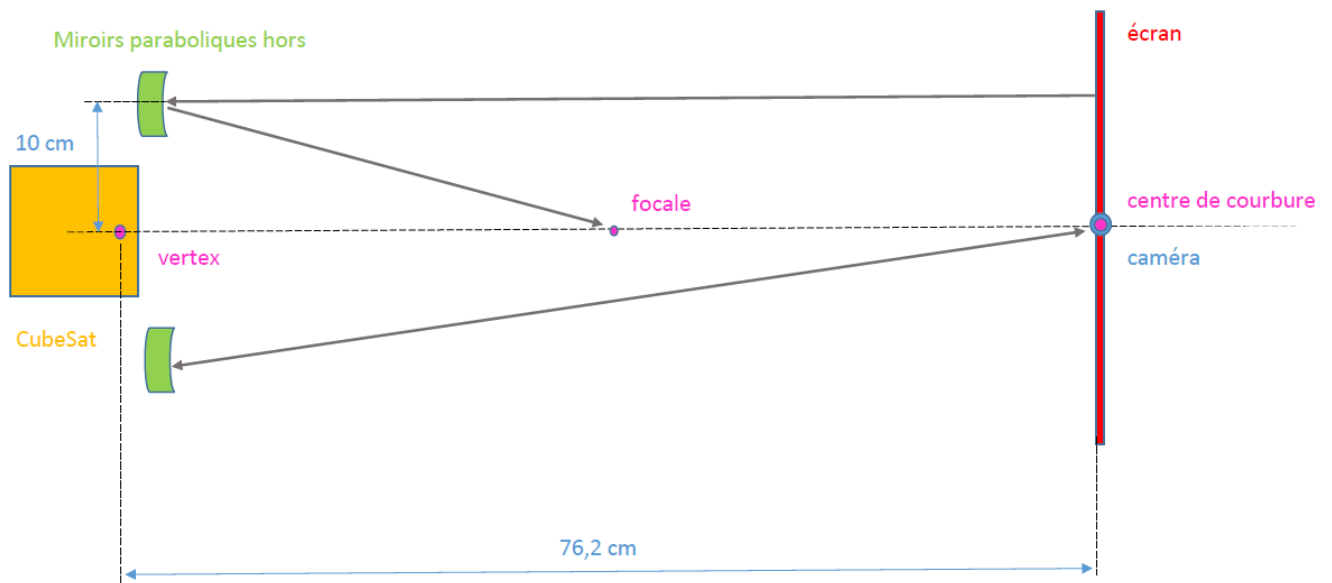


FIGURE 2.10 – Schéma du montage en guise de guide pour la réalisation sur la table optique.

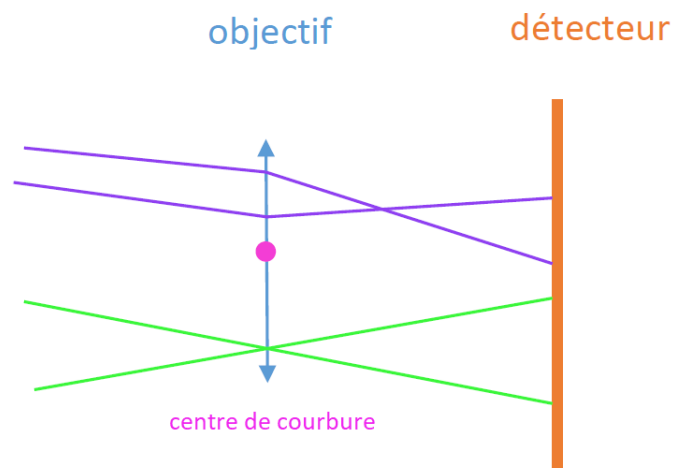


FIGURE 2.11 – Double imagerie pour la configuration choisie. En mauve : utilisation de la lentille objectif pour focaliser sur le miroir. En vert : les rayons issus des franges défocalisées sur le miroir convergent vers le centre de courbure. Ils atteignent le détecteur sans l'intervention de la lentille.

## 2.4 Calibration

Durant la phase de calibration du système, tous les problèmes exposés précédemment ne seront pas pris en compte. Dans un premier temps, seule la non-linéarité de l'écran sera vraiment corrigée. Ce choix est justifié par l'importance de cette erreur dans la mesure. En effet, l'enjeu de la déflectométrie est de déterminer la déflexion de l'image et donc de faire un lien entre les pixels écran et caméra. De plus, la phase de calibration étant lente et très importante, l'idée est de se concentrer sur un problème à la fois. D'autres erreurs seront également réduites comme la saturation, la quantifi-

cation de la fonction sinusoidale. La crise sanitaire causant la fermeture du laboratoire a empêché d'autres études sur la calibration du système telle que la distorsion de la caméra.

Les images sont enregistrées dans un format de fichier non compressé. Le choix se porte ici sur le format BMP (Windows bitmap). De plus, la calibration de l'écran s'effectue dans l'obscurité afin d'éviter toute lumière parasite. En outre, afin d'assurer une bonne réponse des pixels de l'écran, celui-ci doit être allumé 1h à 1h30 avant toutes mesures.

### 2.4.1 Géométrie

La géométrie choisie est de type Hartmann inversé. La distance entre les différents éléments est réalisée grâce aux trous de la table optique. Ces trous sont espacés de 25 mm et servent à fixer différents éléments en les vissant sur la table. La caméra et l'écran se trouvent au centre de courbure de la parabole parente, c'est-à-dire à 76 cm du vertex. De plus, il est préférable de placer l'écran et la caméra sur l'axe optique de la parabole dans le but de réaliser une symétrie dans les résultats entre les miroirs<sup>3</sup>.

Les deux OAPM sont séparés des 20 cm. L'alignement fin entre les deux sera réalisé à l'aide des plaques de translation et des vis de rotation. Le dispositif mis sur pied est illustré à la figure 2.12. Comme expliqué dans l'introduction de ce chapitre, il n'y a qu'un seul miroir à positionner.

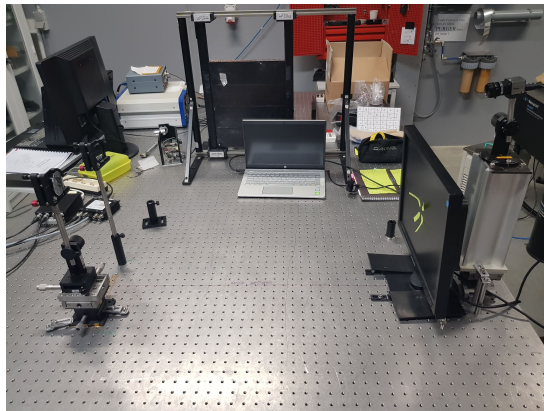


FIGURE 2.12 – Dispositif expérimental. Celui-ci comprend un écran, une caméra et deux montures pour les OAPM.

### 2.4.2 Caméra

Avant toute chose, la caméra doit faire son focus sur le miroir et non sur l'image qu'il renvoie. En effet, le but est de déterminer la position du miroir et non pas de l'image. Cette opération est réalisée tout simplement à l'aide de l'écriture se trouvant sur le support du miroir. Une fois que celle-ci est au point, le miroir l'est aussi. Les paramètres de la caméra sont calibrés à l'aide du logiciel *Thorcam* [49]. Plusieurs sessions de captures d'images furent nécessaires pour notamment ajuster les paramètres de la caméra et de l'écran. Tout d'abord, le comportement des pixels de la caméra est réglé suivant la vidéo d'explications fournie par le fabricant de la caméra[50]. Afin d'éviter la saturation du détecteur, le gain est réglé dans le logiciel avec un facteur 10. Etant donné que la calibration de la caméra est un point très important et fastidieux, dans un premier temps, elle sera considérée comme sans distorsion.

3. Malheureusement, les expériences ont dûes être interrompues à cause du COVID-19.

### 2.4.3 Ecran

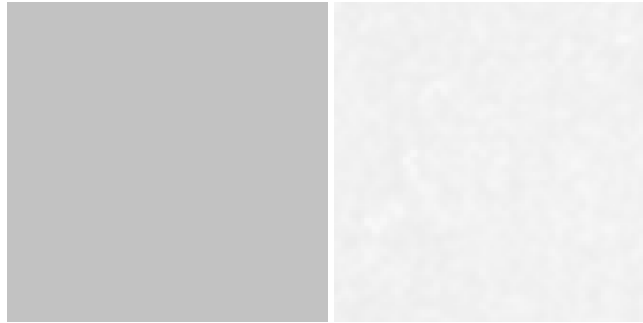
Tout d'abord, il est utile de délimiter la zone de l'écran que le miroir réfléchit vers la caméra. Pour ce faire, une grille de couleur numérotée circule sur tout l'écran et permet de quadriller la zone d'intérêt. Cette zone recouvre plus ou moins 75x75 pixels comme le montre la figure 2.13 avec le quadrillage en post-it.



**FIGURE 2.13** – Quadrillage de la zone de l'écran vue par la caméra au travers du miroir. Cette zone est délimitée par les pointes des post-it. L'utilisation d'un petit écran peut donc être envisagée.

De par cette opération, la miniaturisation peut continuer à être envisagée car un grand écran n'est pas nécessaire pour réaliser l'expérience. Néanmoins, cette observation indique que le système de mesures n'est pas tout à fait au centre de courbure, notamment à cause de l'épaisseur de l'écran.

Comme évoqué dans la section précédente concernant les erreurs systématiques, la réponse de l'écran n'est pas linéaire et possède un certain gamma. Cette partie du travail tentera d'en comprendre les effets et de les résoudre. Durant cette calibration, une succession d'images monochromes dont l'intensité varie de 0 à 255 en niveaux de gris, avec un intervalle régulier de 5, seront affichées. La figure 2.14 illustre la différence entre l'image générée dans le logiciel Matlab et l'image capturée par la caméra. Il est à noter que la luminosité de l'écran est au maximum. Ce choix de luminosité se justifie par le fait que l'angle de vision de la caméra doit être le plus petit possible pour améliorer la précision de mesure. Dès lors, étant donné que le diaphragme de la caméra est très fermé, la luminosité de l'écran doit être élevée et le temps de pause de la caméra être long afin de capter le maximum de photons. Ce long temps de pause de la caméra cause des erreurs liées au bruit thermique.



**FIGURE 2.14** – A gauche : image générée d'intensité 195 en niveau de gris. A droite : image capturée par la caméra de l'image générée de gauche.

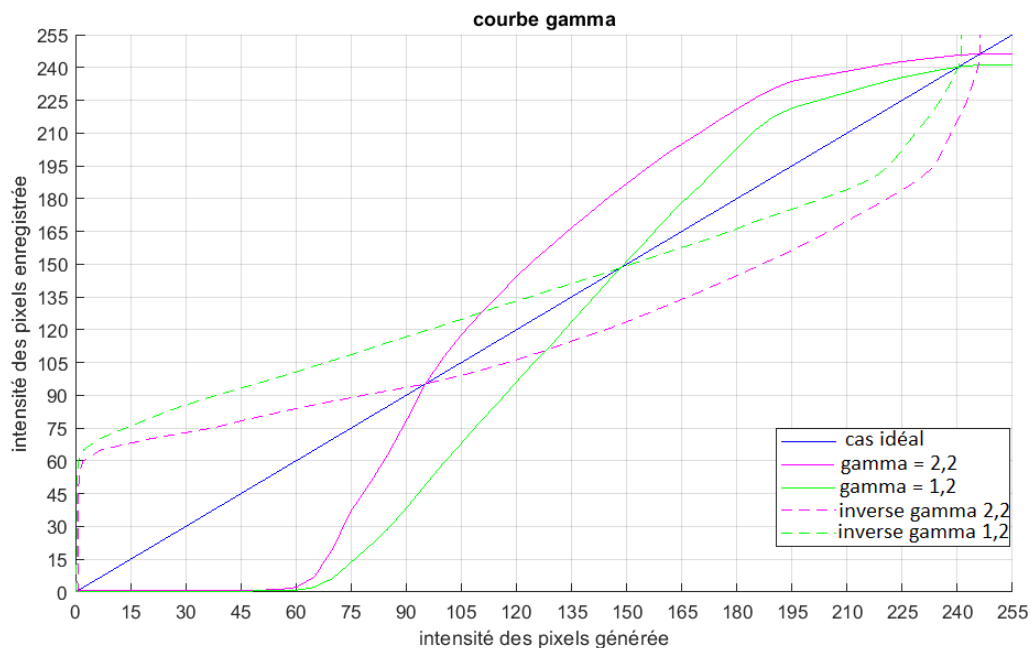
Les images capturées par la caméra ont été traitées avec le logiciel Matlab. Dans une première approximation, la réponse des pixels dans la zone d'intérêt va être considérée comme homogène, c'est-à-dire que tous les pixels vont réagir de la même façon face à l'affichage de l'image. L'intensité de chaque matrice image capturée est recueillie en utilisant la valeur RMS (Root Mean Square en anglais). Cette valeur est calculée comme suit [51]

$$x_{RMS} = \sqrt{\frac{1}{N} \sum_{n=1}^N x_n^2} \quad (2.14)$$

où  $N$  est le nombre d'éléments (pixels) de la matrice et  $x_n$  l'intensité du pixel  $n$ . La présence de pixels morts entache les données. Il faut donc minimiser leur impact en utilisant le calcul RMS.

Sur base de l'introduction faite précédemment sur la valeur gamma d'un écran, une correction gamma de 2, 2 a été appliquée sur la caméra à l'aide du logiciel. En théorie, la réponse du détecteur devrait être linéaire (courbe en bleu), mais il n'en est rien comme le montre la courbe continue en rose sur la figure 2.15. Il ne faut donc pas considérer cette correction comme absolue. Une seconde étude a été menée avec un gamma de 1, 2 appliqué sur la caméra (courbe verte continue). La réponse du détecteur est toujours non linéaire, mais la courbe possède la même tendance que la courbe gamma = 2, 2. Pour effectuer la correction gamma, il suffit d'inverser les axes des abscisses et des ordonnées et d'effectuer un fit des points. Les courbes de correction gamma, appelées courbes de transfert, se trouvent en pointillé sur le graphique 2.15.

La suite de ce travail se basera sur la courbe où un gamma = 2, 2 a été appliqué à la caméra. La démarche à suivre sera identique pour la courbe gamma = 1, 2. La calibration est réalisée en changeant la matrice de couleur du logiciel Matlab. Il s'agit d'une matrice de 256 x 3 éléments. Les trois colonnes sont les composantes RGB d'une couleur. Dans le cas de niveaux de gris, ce triplet sera composé des mêmes valeurs pour un niveau de gris donné. Son fonctionnement est le suivant : si la valeur d'un pixel vaut  $i$  (comprise en 1 et 256) alors celui-ci sera colorié en fonction du code couleur se trouvant à la ligne  $i$  de cette matrice. Pour corriger la non-linéarité de l'écran, la matrice de 256 lignes doit prendre de nouvelles valeurs, celle du fit obtenu sur la courbe inverse de gamma = 2, 2. En d'autres mots, la première ligne cette matrice de couleur de Matlab correspond initialement à l'intensité de gris 0. Dans notre cas, il ne s'agit plus de 0 mais de la valeur du fit et ainsi de suite jusqu'à la ligne 256 qui correspond à la valeur du fit pour une image générée à l'intensité de gris 255.



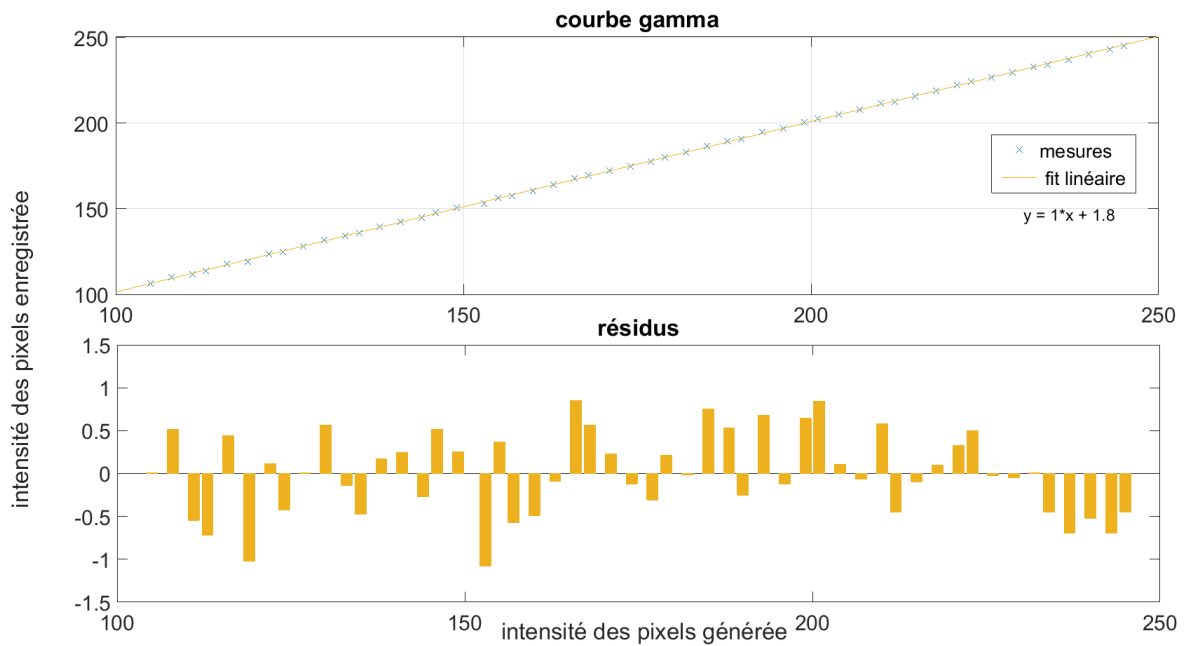
**FIGURE 2.15** – Graphique reprenant l’intensité en niveaux de gris des images capturées en fonction de l’intensité en niveaux de gris des images générées. Deux cas ont été analysés. Le premier en appliquant au préalable un gamma de 2, 2 à la caméra (courbe rose continue) et un deuxième en appliquant un gamma de 1, 2 (courbe verte continue). La droite bleue sur le graphique représente le cas idéal où la réponse de la caméra serait linéaire face à l’intensité affichée sur l’écran. Ce graphique montre l’existence de la non-linéarité de l’écran. Pour corriger la non-linéarité de l’écran, il faut inverser ces courbes continues, obtenant ainsi des courbes de transfert en pointillés.

Pour vérifier que cette nouvelle matrice de couleur contrebalance le gamma de l’écran, une échelle de gris est coloriée avec la nouvelle convention de couleur établie précédemment. Ainsi, les intensités de gris générées par le logiciel avec la nouvelle matrice de couleurs n’auront plus les mêmes teintes de gris qu’avec la matrice de couleur standard comme le montre la figure 2.16.



**FIGURE 2.16** – A gauche : échelle de gris avec la matrice de couleur standard du logiciel. A droite : échelle de gros avec la nouvelle matrice de coloration créée. Les teintes de gris ne sont plus les mêmes.

Grâce aux images capturées, il est d’emblée constaté que la linéarité de la réponse n’est présente que dans une certaine gamme d’intensité. Cet intervalle est compris entre 105 et 245 en niveaux de gris comme le montre la figure 2.17. L’équation de la régression linéaire est  $y = x + 1,8$ . Le graphique **résidus** donne l’écart entre la valeur mesurée et la régression linéaire. Il s’agit de l’erreur commise sur les mesures.



**FIGURE 2.17** – En haut : graphique reprenant l’intensité en niveaux de gris des images capturées en fonction de l’intensité en niveaux de gris des images générées après que l’échelle de gris soit calibrée. Le comportement du système peut être considéré comme linéaire dans la gamme 105 à 245 en niveaux de gris. En bas : erreurs commises sur les mesures.

Dès lors, le 0 de notre échelle (qui est la première ligne de la matrice) sera la valeur du fit à 105 et le 255 (qui est la dernière ligne de la matrice) sera la valeur du fit à 245. Cette façon de construire l’échelle aura comme conséquence de perdre en résolution vu que tous les niveaux de gris ne seront pas utilisés. Cependant, elle rendra n’importe quelle fonction compatible avec la sensibilité en niveaux de gris de notre système.

## Chapitre 3

# Ray tracing

La ligne de conduite suivie durant ce travail fut élaborée sur base de l'état de l'art de la déflectométrie résumé dans le chapitre 1. De plus, le choix de la technique mise en oeuvre fut amplement réconforté par l'article [52] où les auteurs simulent le problème suivant les mêmes étapes choisies. Le but de cette section est donc de poursuivre leur étude sur un CubeSat.

### 3.1 Choix de la technique

L'idée est de partir sur un dispositif le plus simple possible et compactable. Le choix se porte sur une configuration de type test Hartmann inversé. Dans cette méthode, il n'y a que 3 composants : la surface sous test, une caméra et un moniteur. Dans ce mémoire, il faut tenir compte des caractéristiques de la parabole parente dont sont issus les miroirs hors axe. En effet, il est impossible de placer une immense parabole sur un CubeSat. Dès lors, seules des parties de celle-ci seront à bord sur un système déployable pour simuler l'effet de cette parabole. La méthode Hartmann inversé sera néanmoins agrémentée par différentes caractéristiques des autres techniques. En effet, une fois que le montage a été mis en place, il faut déterminer le type d'acquisition des données ainsi que leurs réductions. Ce choix se porte sur le motif sinusoïdal avec plusieurs déphasages. Ainsi, une analyse sur la phase absolue pourra être réalisée. Une approche purement mathématique sur base des polynômes de Zernike permettra de retrouver des informations sur les positions grâce aux valeurs des coefficients.

Cette méthodologie a été réalisée en tenant compte du fait que les miroirs sont considérés comme parfaits. Le but n'est pas d'identifier des défauts locaux mais de positionner au mieux des éléments. La reconstruction 3D de la surface n'est donc pas utile pour ce mémoire. Dès lors, les techniques visant à minimiser les erreurs lors de cette étape ne sont pas considérées, notamment la DPMD.

Comme annoncé au début de ce chapitre, une fois que la méthode fut construite pour essayer de répondre à la question de faisabilité d'alignement de miroirs sur CubeSat, des recherches dans ce sens furent menées et ont conduit à l'article [52]. Ce mémoire tentera donc de poursuivre les recherches réalisées sur une CubeSat.

### 3.2 Algorithme de ray tracing

Le système déflectométrique 3D est simulé à l'aide du logiciel Matlab. L'idée est de partir sur des caractéristiques réelles des miroirs afin de rendre la simulation la plus réaliste possible. Le miroir parabolique hors axe utilisé pour la simulation est décrit dans le chapitre précédent. Néanmoins,

certains raccourcis vont être pris pour étudier un cas simple. La construction du code de ray tracing peut être séparée en plusieurs grandes étapes. Il est à noter que le système déflectométrique sera élaboré en considérant que les rayons lumineux soient issus de la caméra comme évoqué dans le chapitre 1.

### Définition des paramètres de la caméra

Les rayons lumineux sont supposés être issus d'un seul point. Ce point est donc la source de tous les rayons incidents (également appelés rayons sondes dans la littérature). Cette source est située à l'origine des axes et est assimilée à la caméra.

La seconde étape consiste à définir une direction d'observation de la caméra. Celle-ci pointe dans la direction des  $x$  positifs. Une caméra est avant tout caractérisée par son détecteur. La résolution de la caméra est de  $1024 \times 1024$  pixels. Afin de se rapprocher au mieux de la réalité, la dimension d'un pixel de la matrice détecteur est obtenue sur base d'une image prise au laboratoire. Cette taille est calculée en prenant le rapport entre la taille réelle du miroir et sa taille en pixels dans l'image. Une fois cette étape réalisée, le détecteur doit être positionné dans l'espace par rapport au point source. Il ne s'agit pas du détecteur à proprement parler, mais de sa projection dans l'espace. En effet, il est plus facile de travailler avec celle-ci dans l'algorithme afin de définir le sens et la direction des rayons incidents plus aisément.

Ces différents objets mathématiques sont illustrés sur la figure finale du dispositif 3.3. Il est à noter que la matrice de pixels du détecteur n'est pas encore construite.

### Définition des paramètres du miroir parabolique

Le travail se base sur l'utilisation d'un miroir parabolique hors axe. Pour ce faire, il faut d'abord définir la parabole parent 3D (ou parabolôïde) dont il est issu. L'équation générale de celle-ci (sur base de [53]) est

$$x = \frac{-1}{4f}(y^2 + z^2) \quad (3.1)$$

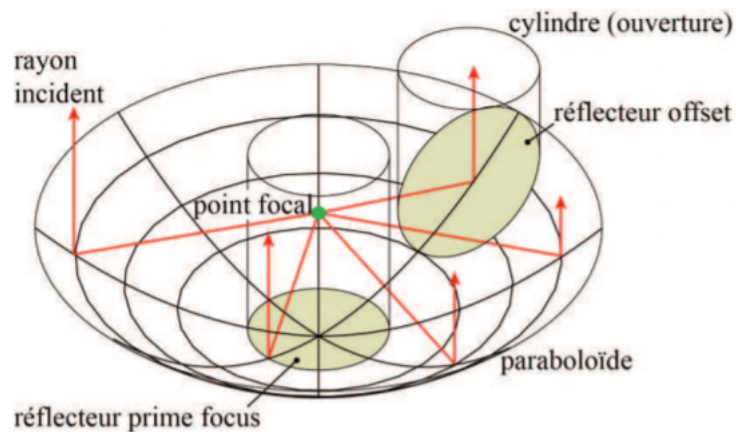
Où  $f$  est la focale du parabolôïde et vaut 381 mm. Elle pointe vers la caméra, c'est-à-dire dans le sens des  $x$  négatifs. L'étape suivante est de positionner cette parabole parent 3D dans l'espace. Pour ce faire, il faut établir les coordonnées du vertex. La distance entre la caméra et le miroir sera conditionnée par la distance entre le point source et le vertex.

### Définition des paramètres du miroir hors axe

Le miroir parabolique hors axe est une portion bien précise du parabolôïde. Les coordonnées du centre de ce miroir sont calculées sur base de caractéristiques réelles. Ce centre est établi avec le  $y$ -offset du miroir parabolique, à savoir 100,32 mm. La coordonnée en  $z$  étant égale à 0, il devient aisé de trouver celle en  $x$  en utilisant l'équation (3.1).

Une fois ce centre défini, un cylindre de rayon de 25 mm est construit autour de celui-ci. L'intersection entre ce cylindre et le parabolôïde donne le lieu géométrique des points équidistants de ce point central et compris dans le rayon du cylindre.

La figure 3.1 illustre ce principe d'intersection entre un cylindre et un parabolôïde pour délimiter le miroir hors axe.



**FIGURE 3.1** – Principe d’obtention d’un miroir parabolique hors axe. Celui-ci est défini par l’intersection entre le paraboloïde et le cylindre centré au centre du miroir hors axe et d’un certain rayon. [54]

### Construction de la grille du détecteur et des rayons incidents

Cette étape n’apparaît que maintenant parce que toute la suite du code se construit pixel par pixel. Une fois que tous les éléments essentiels sont placés dans l’espace, le ray tracing doit commencer par l’élaboration des rayons incidents. Chaque rayon est issu du point source et passe dans un pixel du détecteur de la caméra pour atteindre un point du miroir. La construction de la grille projetée du détecteur est donc une étape cruciale. Pour ce faire, on définit tout d’abord le centre et les axes de celle-ci. Tous les autres pixels sont construits autour de ce centre en respectant bien la résolution de l’écran, ainsi que la taille des pixels calculée précédemment.

Les rayons incidents sont construits comme des droites dont l’origine est le point source et passant par un pixel (un pixel pour un rayon) du détecteur projeté.

Le tout est affiché dans l’espace afin de vérifier la construction de ces éléments. Il est à noter que le nombre de pixels du détecteur, ainsi que de rayons, est considérablement réduit dans un premier temps afin de faciliter la visualisation 3D du système.

### Intersection rayon incident-miroir

L’idée est de calculer le point d’intersection entre le rayon incident et le miroir parabolique hors axe. Cette étape est très importante car tout le processus de réflexion du rayon se construit autour de ce point. Il faut donc résoudre l’équation de la droite qui définit le rayon incident dans l’espace 3D avec l’équation du paraboloïde. Seuls les points se trouvant dans la partie hors axe souhaité sont conservés pour la suite du programme. Les points sont ensuite affichés afin de vérifier qu’il s’agisse bien de l’intersection entre le rayon incident et le miroir parabolique hors axes. De plus, ils satisfont aux équations.

### Calcul de la normale au point d’intersection

La normale à la surface, au point de contact calculé précédemment, s’obtient de manière analytique, c’est-à-dire en prenant le gradient [55] de la fonction analytique du paraboloïde et non à partir de la surface construite dans le logiciel.

$$n = \nabla F(x, y, z) \quad (3.2)$$

Où  $n$  est la normale au point  $(x, y, z)$  et  $F$  est la fonction décrivant le paraboloïde, c'est-à-dire  $F(x, y, z) = \frac{-1}{4f}y^2 + z^2 - x$ . Cette opération permet de récupérer les composantes de la normale.

Afin de vérifier l'algorithme, le vecteur normal est affiché au point de contact confirmant le sens et la direction de celui-ci. Une surface plane est simulée pour vérifier que le calcul donne bien les composantes des normales parallèles entre elles.

### Calcul du rayon réfléchi au point d'intersection

Le calcul du rayon réfléchi n'est pas trivial en informatique. Pour ce faire, les lois de la géométrie vectorielle rentrent en application [56]. L'analyse vectorielle ci-dessous est représentée à la figure 3.2. Dans un premier temps, le vecteur directeur du rayon incident est inversé car son sens est important. En effet, il doit avoir le même sens que le vecteur normal. Cette opération est réalisée afin de connaître l'angle orienté entre la normale et le rayon incident. Etant donné que ces deux vecteurs sont non nuls, l'angle  $\theta$  est [56]

$$\cos(\theta) = \frac{\langle \mathbf{i}, \mathbf{n} \rangle}{|\mathbf{i}| \cdot |\mathbf{n}|} \quad (3.3)$$

Où  $\theta$  est l'angle orienté  $\in [0, 2\pi[$  entre les deux vecteurs  $\mathbf{i}$  (incident) et  $\mathbf{n}$  (normal). Afin de calculer l'angle dans le bon quadrant, il faut passer par la fonction tangente dont la définition est le rapport entre la fonction sinus et la fonction cosinus. L'angle entre deux vecteurs est également donné par [56]

$$|\mathbf{i} \wedge \mathbf{n}| = |\mathbf{i}| \cdot |\mathbf{n}| \cdot \sin(\theta) \quad (3.4)$$

Dès lors, l'expression pour retrouver  $\theta$  est

$$\begin{aligned} \tan(\theta) &= \frac{\sin(\theta)}{\cos(\theta)} \\ &= \frac{|\mathbf{i} \wedge \mathbf{n}|}{\langle \mathbf{i}, \mathbf{n} \rangle} \end{aligned} \quad (3.5)$$

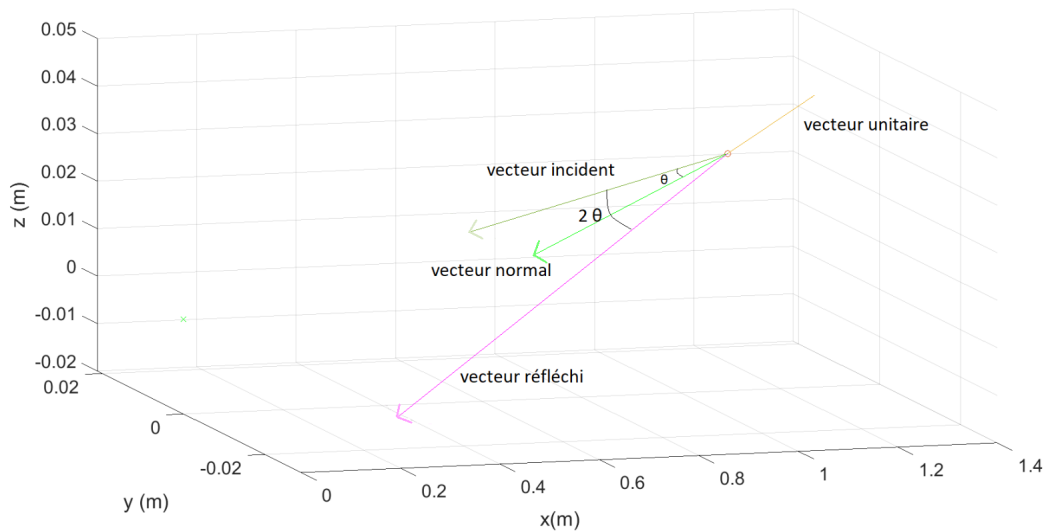
La fonction utilisée sur Matlab est `atan2` donnant comme intervalle pour l'angle  $[-\pi, \pi[$ .

Une fois que l'angle est récupéré, l'idée est de faire tourner le vecteur incident de  $2\theta$  autour d'un axe de rotation pour obtenir le vecteur réfléchi. Cet axe n'est rien d'autre qu'un vecteur unitaire perpendiculaire au plan formé par le vecteur incident et le vecteur normal. La rotation s'effectue en définissant la matrice de rotation [57]  $R =$

$$\begin{pmatrix} \cos(\theta) + u_x^2(1 - \cos(\theta)) & u_x u_y(1 - \cos(\theta)) - u_z \sin(\theta) & u_x u_z(1 - \cos(\theta)) + u_y \sin(\theta) \\ u_y u_x(1 - \cos(\theta)) + u_z \sin(\theta) & \cos(\theta) + u_y^2(1 - \cos(\theta)) & u_y u_z(1 - \cos(\theta)) - u_x \sin(\theta) \\ u_z u_x(1 - \cos(\theta)) - u_y \sin(\theta) & u_z u_y(1 - \cos(\theta)) + u_x \sin(\theta) & \cos(\theta) + u_z^2(1 - \cos(\theta)) \end{pmatrix}$$

Où  $(u_x, u_y, u_z)$  sont les composantes du vecteur unitaire et  $\theta$  (en rad) l'angle entre le vecteur incident et normal. Cette matrice est appliquée sur le vecteur incident pour donner le vecteur réfléchi. De par cette opération, les trois vecteurs (incident, normal et réfléchi) sont dans le même plan.

Cette étape de l'algorithme est validée en comparant les angles incidents et réfléchis. La précision de la machine devient connue. En effet, les deux angles sont égaux à  $10^{-71}$  rad près.



**FIGURE 3.2** – Le vecteur incident effectue une rotation de  $2\theta$  ( $\theta$  étant l'angle entre le vecteur incident et le vecteur normal) autour d'un axe perpendiculaire unitaire pour donner le vecteur réfléchi.

### Intersection rayon réfléchi-écran

L'écran de  $1024 \times 1024$  pixels et de  $30 \times 30$  cm n'est rien d'autre qu'un plan dans un espace 3D dont sa position est tout d'abord définie. Cette étape suit la même logique que le calcul d'intersection du rayon incident avec le paraboloïde. La seule différence est qu'ici le point de départ se trouve sur ce paraboloïde et le vecteur directeur utilisé est celui du rayon réfléchi obtenu dans l'étape précédente.

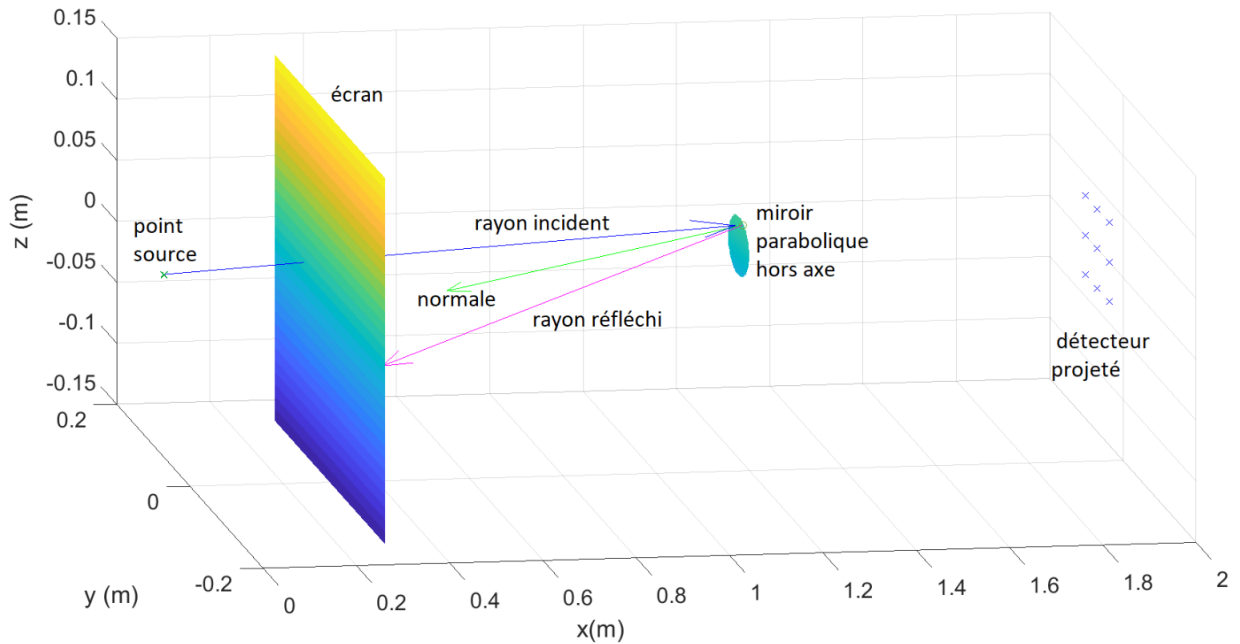
### Création d'un signal cosinus et coloration en gris

L'écran affiche des franges cosinus qui vont être colorées en niveaux de gris. La fonction doit donc osciller entre la valeur 0 à 255 sur plusieurs périodes.

### Association des valeurs de gris de l'écran aux pixels du détecteur

La matrice détecteur doit maintenant être remplie en niveaux de gris également. Elle permet de récupérer ce que voit la caméra sur le miroir, c'est-à-dire le motif à franges réfléchi. Pour ce faire, une analyse du parcours des rayons est nécessaire. Si un rayon incident, passant par le pixel  $(i, j)$  de la matrice détecteur, est réfléchi par le miroir sur le pixel écran  $(a, b)$ , alors le pixel  $(i, j)$  se verra attribué le même niveau de gris que le pixel écran  $(a, b)$ . Les rayons ne faisant pas partie du processus de réflexion se verront attribués la couleur noire. Ce choix est justifié par le fait qu'en laboratoire, les images ont été prises dans l'obscurité pour éviter de la lumière parasite. Dès lors, seul le miroir était illuminé et l'environnement était noir.

L'algorithme de ray tracing pour un système de déflectométrie 3D étant conçu, le résultat final est illustré à la figure 3.3. On y retrouve le point source (ou la caméra), l'écran, le miroir parabolique hors axe, la matrice de pixel détecteur projeté, ainsi que le rayon incident et réfléchi avec la normale dans le même plan.



**FIGURE 3.3** – Visualisation 3D du système de déflectométrie avec les différents éléments intervenant dans le processus. Le rayon incident issu du point source se réfléchit sur le miroir parabolique hors axe. Le rayon réfléchi est calculé par rapport à la normale à la surface et vient toucher un pixel de l'écran.

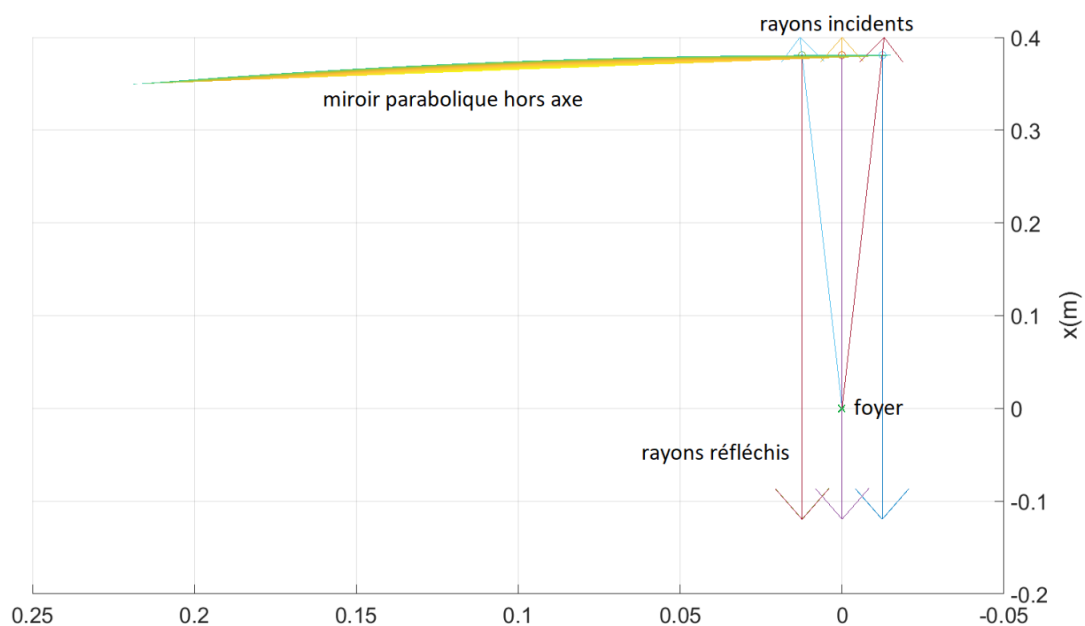
### Vérification de système dans son ensemble

Une fois que l'ensemble du système a été construit dans l'algorithme, son bon fonctionnement physique doit être testé avant de passer à l'étude de notre cas. Pour valider l'algorithme, le système simulé doit satisfaire à quelques propriétés d'optique géométrique [58]. Ces principes restent vrais quel que soit le type de miroir hors axe. En effet, Le hors axe reste quand même un morceau du paraboloïde. Il doit donc satisfaire aussi ses propriétés.

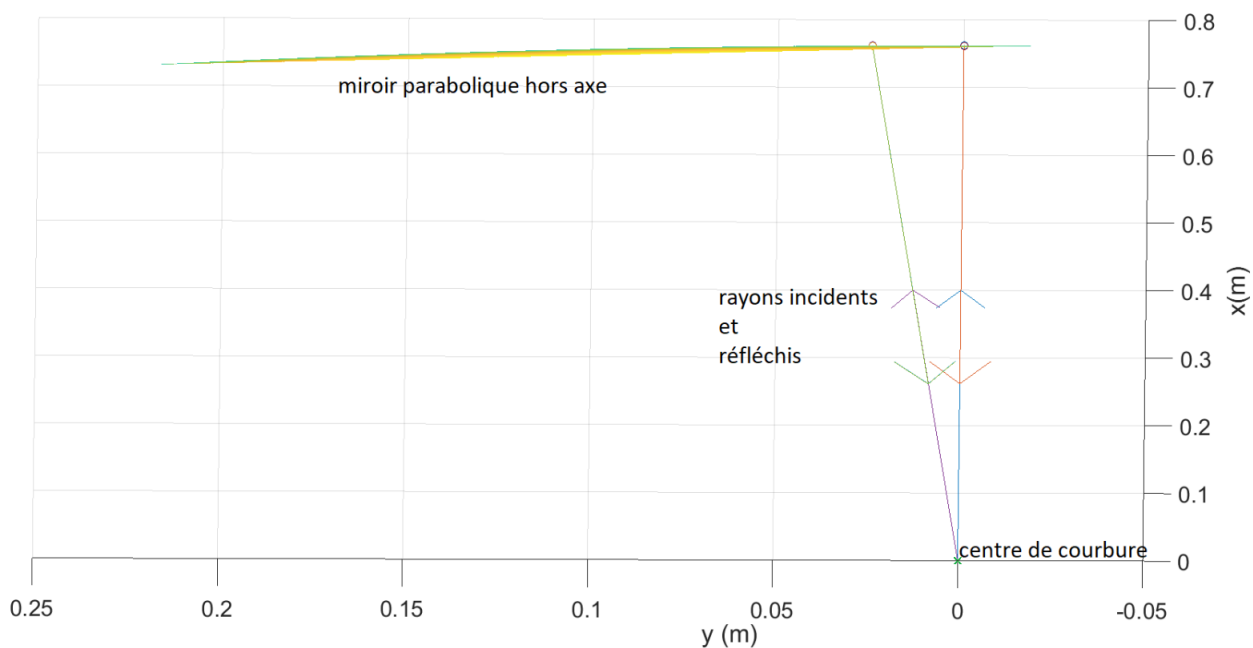
La première peut être formulée de la sorte : tout rayon issu du foyer d'une parabole sera dévié par celle-ci parallèlement à l'axe optique. Il est important de garder à l'esprit que la parabole en question est la parabole parent 3D du système. La figure 3.4 simule cette propriété avec succès. Le point source étant placé au foyer du paraboloïde, soit à 381 mm du vertex, les rayons réfléchis sont en effet parallèles à l'axe optique et entre eux.

La deuxième propriété est que si les rayons incidents sont issus du centre de courbure du paraboloïde, alors ils sont réfléchis en ce point. De nouveau, cette propriété est parfaitement illustrée à la figure 3.5. Dans notre cas, le centre de courbure se trouve à  $2 \cdot \text{focale} = 762$  mm.

La trajecte des rayons lumineux est précis à  $10^{-6}$  m ce qui impactera le domaine d'étude des translations. En effet, le désalignement va être étudié d'abord un mm et puis en centaine de  $\mu\text{m}$ . Plus bas, les algorithmes commettront trop d'erreur pour observer quoi que ce soit. Ainsi, toutes ces vérifications attestent que l'algorithme est fiable et est physiquement correct.



**FIGURE 3.4** – Simulation vue d'en haut validant un principe d'optique géométrique. Les rayons issus du foyer d'un paraboloïde sont réfléchis vers l'infini parallèlement entre eux et à l'axe optique.



**FIGURE 3.5** – Simulation vue d'en haut validant un principe d'optique géométrique. Les rayons lumineux partent du centre de courbure d'une paraboloïde et sont réfléchis sur celui-ci.

## Remarques

Durant les premières prises de mesures, l'image fournie par le détecteur présentait des escaliers dans les franges, mais aussi sur les bords du miroir censés être courbés comme le montre la figure

3.6. Ce phénomène, fort connu en informatique, s'appelle le crénelage ou aliasing en anglais [59]. Cet effet escalier traduit le fait que le motif cosinus codé sur l'écran, ainsi que le miroir ont perdu



**FIGURE 3.6** – Image du miroir avec les franges données par le détecteur de la caméra. Des figures en escalier sont présentes un peu partout. C'est le phénomène de crénelage.

leur continuité. En effet, le cosinus est échantillonné sur un écran de 1024 x 1024 pixels. Dès lors, le motif observé sur le miroir courbé présente des escaliers [59]. L'effet crénelage dépend de la taille des pixels, mais aussi de la différence de valeurs entre deux pixels voisins. En effet, l'escalier est d'autant plus perceptible que le saut de valeur entre deux pixels est grand, comme passer du noir au blanc. Il est impossible d'avoir un résultat entre deux pixels.

Pour contourner ce problème, l'écran va être surpixelisé de manière à lisser au maximum la courbe cosinus. Ainsi, l'écran aura une résolution de 10240 x 10240 pixels.

Le temps de calcul de l'ordinateur est fort important lorsque la résolution de la caméra est de 1024 x 1024 pixels. Afin de réduire ce temps, quelques modifications doivent être apportées. Elles ne sont pas sans conséquence sur la précision du résultat, mais c'est un choix à faire au vu que la crise sanitaire empêche tout accès à un ordinateur de calcul adapté. Dès lors, la résolution de la caméra est nettement réduite. La matrice de pixels détecteur est parcouru par pas de 8 sur ses 1024 x 1024 pixels. Afin de visualiser tout le miroir durant les prises de mesures, la caméra sera déplacée de (0, 0, 0) à (0, 0.1, 0). Le motif affiché sur l'écran possède 40 périodes et sera à la position de 0, 4 m du point source. Dans une première approximation, l'écran sera devant la caméra et ne bloquera pas les rayons issus de celle-ci.

### 3.3 Algorithmes de traitement des données

Une fois que les images des franges cosinus sont enregistrées, elles doivent être traitées et analysées pour en retirer les informations souhaitées. Les quatre images déphasées de  $\frac{\pi}{2}$  dans les deux directions de modulation sont traitées par différents algorithmes dans Matlab.

#### 3.3.1 Enroulage de la phase

La première étape du traitement des données est l'enroulage de phase dont le but est d'extraire la phase enroulée de chaque pixel directement à partir des images. L'intensité capturée pour chaque

pixel est donnée par [31]

$$I_k(y, z) = A + B.\cos(\phi_{enr} + \frac{2k\pi}{2}) \quad (3.6)$$

où  $y$  et  $z$  sont les coordonnées du pixel écran,  $k$  est le numéro de l'image,  $A$  est l'intensité de fond et  $B$  est l'amplitude du signal. Pour chaque direction de franges (horizontales ou verticales), quatre images de franges sont capturées donnant ainsi comme intensité pour chaque pixel

$$I_0(y, z) = A + B.\cos(\phi_{enr} + 0) = A + B.\cos(\phi_{enr}) \quad (3.7)$$

$$I_1(y, z) = A + B.\cos(\phi_{enr} + \frac{\pi}{2}) = A + B.\sin(\phi_{enr}) \quad (3.8)$$

$$I_2(y, z) = A + B.\cos(\phi_{enr} + \pi) = A - B.\cos(\phi_{enr}) \quad (3.9)$$

$$I_3(y, z) = A + B.\cos(\phi_{enr} + \frac{3\pi}{2}) = A - B.\sin(\phi_{enr}) \quad (3.10)$$

La phase enroulée  $\phi_{enr}$  est déterminée à partir de l'acquisition de ces 4 images par la relation

$$\tan(\phi_{enr}) = \frac{I_1 - I_3}{I_0 - I_2} = \frac{\sin(\phi_{enr})}{\cos(\phi_{enr})} \quad (3.11)$$

On recommence l'opération pour les franges dans la direction perpendiculaire pour au final obtenir deux cartes de phase enroulée.

### 3.3.2 Création d'un masque

Le déroulage de phase ne doit s'effectuer que sur le miroir. Dès lors, un masque doit être appliqué sur les cartes de phases enroulées avant l'opération de déroulage de celles-ci. Pour ce faire, toutes les 8 images du détecteur sont additionnées entre elles. Une valeur seuil est ainsi déterminée pour considérer tous les pixels en dessous de ce nombre comme étant des NaN (Not a Number en anglais). Ainsi, les parties hors miroir ne seront pas traitées par les algorithmes ci-dessous.

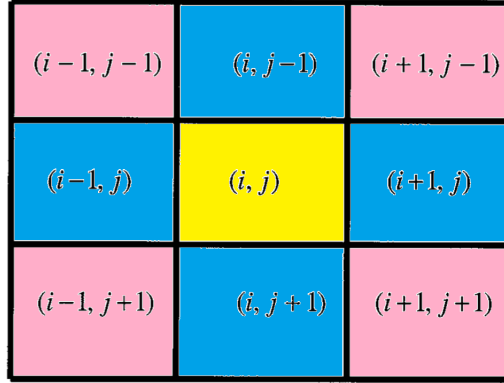
### 3.3.3 Déroulage de la phase

Le déroulage de phase est l'étape la plus délicate et la moins triviale. Il existe de nombreuses études et algorithmes pour tenter de dérouler la phase avec le moins d'erreurs possible. Le déroulage peut être local comme global, en utilisant les moindres carrés ou encore sur base de modèles [60].

L'algorithme choisi pour ce mémoire est l'algorithme d'Herraez proposé dans l'article [61] et trouvé sur [62]. Il s'agit d'un déroulage local de phase guidé sur un chemin de pixel. Ce type d'algorithmes possède deux grandes étapes. En effet, pour obtenir un déroulage de phase de qualité, il faut trouver une fonction de fiabilité et un chemin de déroulage minimisant les erreurs (les erreurs dépendent bien évidemment du chemin emprunté). Ainsi, le but de l'algorithme est de déterminer la qualité de chaque pixel et, en fonction de leur fiabilité, composer un chemin de déroulage. Contrairement à d'autres algorithmes de déroulage sur un chemin, le déroulage décrit ci-dessous suit un chemin discret.

### Fonction de fiabilité

Une fonction de fiabilité est une opération mathématique donnant à chaque pixel un niveau de confiance pour participer au déroulage de phase sans provoquer des erreurs. Pour ce faire, la méthode appelée la double différence est appliquée à chaque pixel. Le principe illustré à la figure 3.7 est expliqué ci-après. L'évaluation de la fiabilité d'un pixel à la position  $(i, j)$  (en jaune) est réalisée en faisant participer les pixels voisins orthogonaux (en bleu) et diagonaux (en rose). Cette fonction travaille donc dans une sous-matrice de  $3 \times 3$  pixels.



**FIGURE 3.7** – Sous-matrice de pixels intervenant dans le calcul de fiabilité du pixel  $(i, j)$ . Il s'agit de la méthode dite de la double différence qui fait entrer en jeu les pixels voisins orthogonaux (en bleu) et diagonaux (en rose). [61]

L'expression de cette différence pour un pixel  $(i, j)$  est [61]

$$D(i, j) = [H^2(i, j) + V^2(i, j) + D_1^2(i, j) + D_2^2(i, j)]^{1/2} \quad (3.12)$$

Avec

$$\begin{aligned} H(i, j) &= \gamma[\varphi(i-1, j) - \varphi(i, j)] - \gamma[\varphi(i, j) - \varphi(i+1, j)] \\ V(i, j) &= \gamma[\varphi(i, j-1) - \varphi(i, j)] - \gamma[\varphi(i, j) - \varphi(i, j+1)] \\ D_1(i, j) &= \gamma[\varphi(i-1, j-1) - \varphi(i, j)] - \gamma[\varphi(i, j) - \varphi(i+1, j+1)] \\ D_2(i, j) &= \gamma[\varphi(i-1, j+1) - \varphi(i, j)] - \gamma[\varphi(i, j) - \varphi(i+1, j-1)] \end{aligned} \quad (3.13)$$

Où  $\gamma$  est un opérateur qui supprime les sauts de  $2\pi$  entre deux pixels voisins. Les pixels les plus fiables sont ceux qui ont une double différence le plus faible. Ainsi, le facteur de qualité  $R$  est défini comme [61]

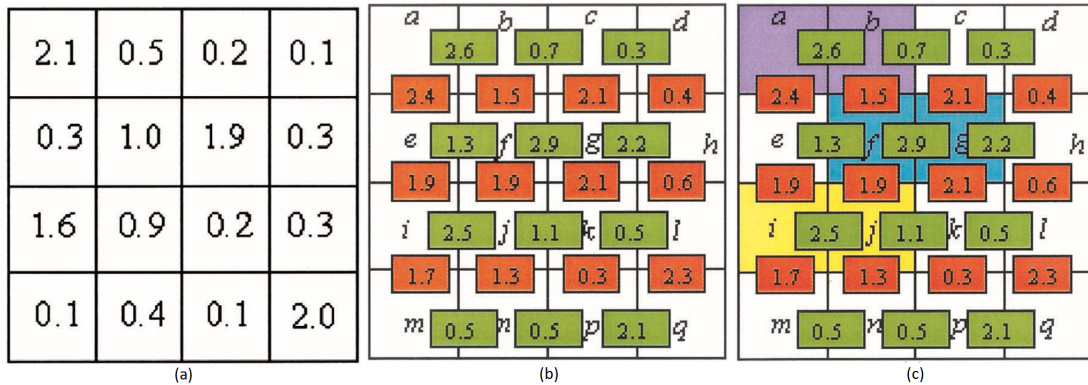
$$R = \frac{1}{D} \quad (3.14)$$

### Choix du chemin de déroulage

Tout le processus de déroulage de phase sera illustré par l'exemple des figures 3.8, 3.9 et 3.10. La première étape consiste à calculer la fiabilité de chaque pixel de l'image avec la fonction de fiabilité décrite précédemment. Ainsi, chaque pixel est associé à une valeur (figure 3.8a). Le chemin suivi pour le déroulage de phase n'est pas exactement déterminé directement par la fiabilité du pixel, mais plutôt par la fiabilité des bords de celui-ci. En effet, un pixel partage avec son voisin orthogonal, une

arête verticale ou horizontale. Le calcul de la fiabilité des bords d'un pixel s'obtient en additionnant les fiabilités des deux pixels voisins formant un bord (figure 3.8b).

Le chemin de déroulage commence à l'arête ayant la valeur la plus élevée et se termine sur l'arête ayant la moins bonne fiabilité. Sur la figure 3.8c, le chemin commence donc sur l'arête ayant la valeur 2.9. Les pixels  $f$  et  $g$  sont déroulés l'un par rapport à l'autre formant ainsi le groupe déroulé turquoise. Le deuxième groupe est le groupe mauve, ainsi de suite. Ainsi, une succession de groupes de 2 pixels sont d'abord construits.



**FIGURE 3.8** – Exemple d'un déroulage de phase suivant un chemin de qualité : (a) valeur de fiabilité pour chaque pixel. (b) fiabilité de chaque arête permettant ainsi de construire le chemin de déroulage. En effet, celui-ci suit un ordre décroissant de fiabilité d'arêtes. (c) construction des premiers groupes déroulés. [61]

Le déroulage d'un pixel par rapport à un autre s'effectue à l'aide d'un opérateur  $U$  [63]. Il permet de connaître le multiple entier de  $2\pi$  à ajouter ou soustraire au pixel qui est en train d'être déroulé

$$U(x_i, x_j) = 2\pi \left\lfloor \frac{x_i - x_j + \pi}{2\pi} \right\rfloor \quad (3.15)$$

Où  $\lfloor \cdot \rfloor$  est la fonction plancher,  $x_i$  est la valeur du pixel à dérouler par rapport à la valeur du pixel  $x_j$ . Il s'agit d'une fonction qui rend le plus grand entier inférieur à son argument.

Le déroulage de phase doit respecter la condition d'Itoh. Celle-ci s'exprime par la relation [64]

$$|\phi_u(n) - \phi_u(n - 1)| < \pi \quad (3.16)$$

Où  $\phi_u$  est la phase déroulée et  $n$  le numéro du pixel. Ainsi, la phase déroulée de deux pixels voisins ne peut pas varier brutalement. En effet, la correction de  $2\pi$  pour supprimer les discontinuités présentes dans la phase enroulée ne lisse pas toujours parfaitement la droite. Des sauts brusques de valeurs peuvent encore être présents.

L'étape suivante est de dérouler un pixel sans groupe par rapport à un autre pixel appartenant déjà à un groupe grâce à l'opérateur  $U$ . Il va de soi que l'ordre de déroulage continue à suivre un ordre décroissant dans la fiabilité des arêtes. Suite à cela, le groupe s'agrandit avec le pixel qui vient d'être déroulé comme le montre la figure 3.9d où le pixel  $e$  est déroulé par rapport au pixel  $a$ . Il rejoint dès lors le groupe mauve.

La dernière étape est le déroulage d'un groupe par rapport à un autre (figures 3.10 e, f et g). En effet, étant donné que les pixels ont été déroulés localement et de façon discontinue, le but est de les réunir pour ne former plus qu'un seul groupe. Dès lors, le pixel du plus petit groupe formant l'arête est déroulé par rapport à l'autre pixel formant l'arête du plus grand groupe. Le multiple de  $2\pi$  ajouté

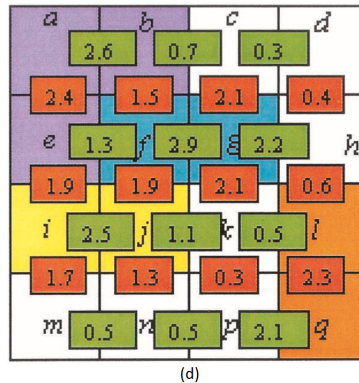


FIGURE 3.9 – Déroulage du pixel  $e$  n'appartenant à aucun groupe par rapport au pixel  $a$  appartenant au groupe de déroulage mauve. Le pixel  $e$  intègre ce groupe à la fin de l'opération. [61]

ou soustrait se fait sur tous les pixels du plus petit groupe. Cela revient à dire que le plus petit groupe est déroulé par rapport au plus grand groupe. A la fin de l'opération, les deux groupes ont fusionné en un et gardent les couleurs du groupe dominant (passage de la figure 3.9d à la figure 3.10e). Si les deux groupes ont le même nombre de pixels comme sur l'image 3.10e avec le groupe mauve et le groupe turquoise, le groupe dominant est considéré comme le groupe ayant les arêtes de fiabilité les plus élevées avec les pixels non déroulés. Dans l'exemple, il s'agit du groupe mauve qui possède une arête à 1.7 avec le pixel non déroulé  $m$ .

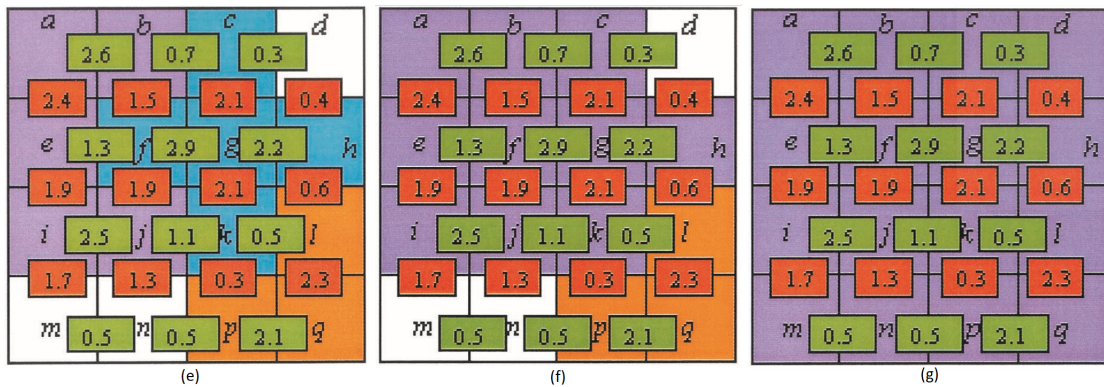


FIGURE 3.10 – Déroulage de groupes de pixels déroulés. Le groupe de plus faibles pixels est déroulé par rapport au groupe de pixels dominants (de (e) vers (g)). [61]

Les bords de la matrice de pixels sont finalement déroulés par rapport au groupe final contenant les autres pixels déroulés. Il est à noter que les pixels ayant une trop faible fiabilité peuvent ne pas participer au déroulage ou peuvent être interpolés à la toute fin du déroulage.

Les avantages de cet algorithme sont multiples [61] [63]. Premièrement, la rapidité dans le déroulage. En effet, l'opération s'effectue dans l'ordre de la seconde. De plus, le chemin suivi évite la propagation des erreurs tout le long du déroulage puisque les pixels ayant une moins bonne fiabilité sont déroulés en toute fin. Cela a pour avantage d'isoler les mauvaises données.

### 3.3.4 Polynômes de Zernike

#### Généralité

Inventés par Frédéric Zernike vers 1934, ces polynômes reconstruisent la forme du front d'onde après son passage dans une ouverture circulaire [65]. Le but de ces polynômes est de quantifier les aberrations optiques indépendamment les unes des autres en pondérant les polynômes par des coefficients de Zernike  $C_n^m$  [66]. En coordonnées polaires, ces polynômes s'expriment de manière générale sur un disque de rayon unité. Il existe plusieurs expressions mathématiques des polynômes de Zernike suivant la notation des indices. La notation de James C. Wyant se trouve ci-dessous [67].

$$Z_n^{-m}(r, \theta) = R_n^m(r) \cos(m\theta) \quad (3.17)$$

$$Z_n^m(r, \theta) = R_n^m(r) \sin(m\theta) \quad (3.18)$$

Où  $r$  est la distance par rapport à l'axe optique,  $m$  (fréquence angulaire) et  $n$  (ordre radial) sont des entiers positifs avec  $0 \leq m \leq n$ , et  $0 \leq r \leq 1$ . L'angle azimutal  $\theta$  est exprimé en radian sur le cercle unitaire. La fonction radiale est définie par [66]

$$R_n^m(r) = \sum_{l=0}^{(n-m)/2} \frac{(-1)^l (n-l)!}{l! [\frac{1}{2}(n+m)-l]! [\frac{1}{2}(n-m)-l]!} r^{n-2l} \quad (3.19)$$

si  $(n-m)$  est pair et est nul si  $(n-m)$  est impair. Ce modèle renferme donc 2 types de polynômes orthogonaux suivant la parité. Voici un tableau reprenant les 10 premiers polynômes de Zernike avec leur signification [67].

polynôme	valeur	signification
$Z_0^0(r, \theta)$	1	piston
$Z_1^{-1}(r, \theta)$	$r \cos(\theta)$	x-tilt
$Z_1^1(r, \theta)$	$r \sin(\theta)$	y-tilt
$Z_2^{-2}(r, \theta)$	$r^2 \cos(2\theta)$	astigmatisme à 0 °
$Z_2^0(r, \theta)$	$2r^2 - 1$	defocus
$Z_2^2(r, \theta)$	$r^2 \sin(2\theta)$	astigmatisme à 45 °
$Z_3^{-3}(r, \theta)$	$r^3 \cos(3\theta)$	trefoil sur x
$Z_3^{-1}(r, \theta)$	$(3r^3 - 2r) \cos(\theta)$	coma sur x
$Z_3^1(r, \theta)$	$(3r^3 - 2r) \sin(\theta)$	coma sur y
$Z_3^3(r, \theta)$	$r^3 \sin(3\theta)$	trefoil sur y

Toute fonction peut se décomposer en une somme de polynômes de Zernike pondérée par des coefficients  $C_n^m$  [67].

$$f(r, \theta) = \sum_{n=0}^{\infty} \sum_{m=-n}^n C_n^m Z_n^m(r, \theta) \quad (3.20)$$

$$= \sum_i C_i Z_i \quad (3.21)$$

où  $f(r, \theta)$  est une fonction connue. La valeur des coefficients donne l'écart du miroir par rapport à une sphère parfaite.

Dans ce mémoire, ce sont les erreurs de type piston, x-tilt (ou tip) et y-tilt qui seront traitées, c'est-à-dire des erreurs d'alignement. Leur effet est illustrée à la figure 3.11. Ces trois termes ne sont pas considérés comme des aberrations de l'optique.

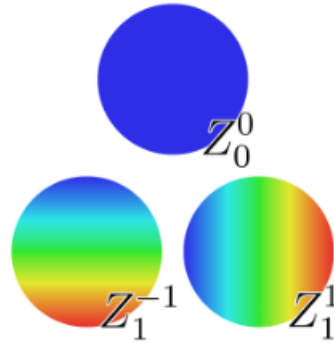


FIGURE 3.11 – Les trois premiers modes de Zernike : piston (en haut), tip (à droite), tilt (à gauche). [68]

Le piston représente un décalage en hauteur par rapport à la surface de référence. Le tip et le tilt sont des inclinaisons en x et en y par rapport à la référence.

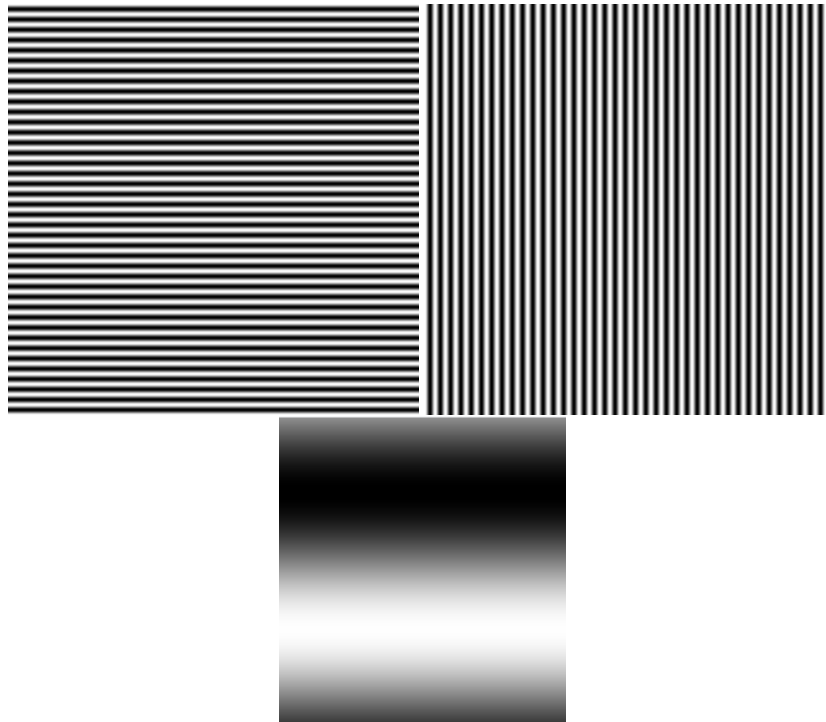
### Application

Dans ce mémoire, l'idée est de faire un fit des cartes de phase absolue par les polynômes de Zernike. Cependant, il est à noter que la signification physique des termes de Zernike donnée précédemment n'a lieu que pour un fit sur une carte de coordonnées. Ce travail doit donc d'abord déterminer quels coefficients des polynômes varient le plus significativement lors de translations ou rotations. En d'autres mots, l'enjeu est de déterminer la nouvelle signification des termes sur une carte de phase. Dans ce mémoire, les fonctions de Zernike sont obtenues à l'aide de la fonction zernfun [69].

## 3.4 Alignement parfait

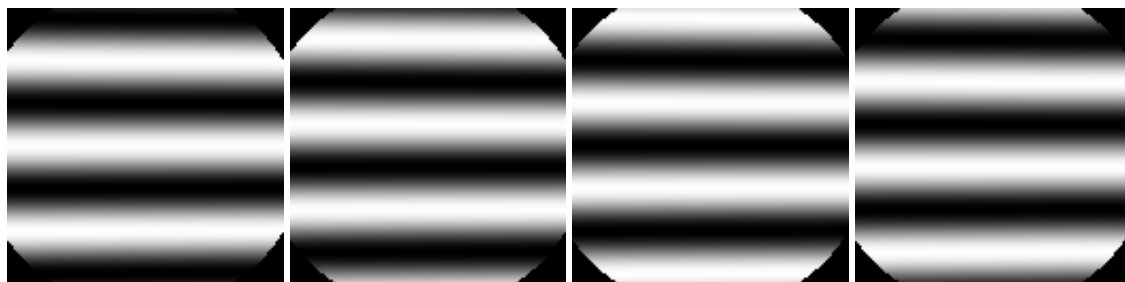
Dans cette section, le miroir hors axe est supposé être dans un alignement parfait. Les cartes de phase déroulées sont analysées à l'aide des polynômes de Zernike pour en ressortir des coefficients. En effet, l'alignement parfait dans la configuration choisie possède une certaine signature dans les coefficients. Le but de cette section est d'en retirer les valeurs et d'étudier leur évolution lors des désalignements. Les coefficients dans la configuration parfaite vont donc être considérés comme étant des offsets, c'est-à-dire qu'ils vont être remis à 0 afin d'étudier leur sensibilité face aux désalignements.

Pour étudier le système, l'écran affiche une succession d'images de franges déphasées de  $\frac{\pi}{2}$  horizontalement et verticalement. La figure 3.12 montre une image de franges horizontales et de franges verticales utilisées. La troisième image est un zoom sur une période afin de bien distinguer le dénivelé de niveaux de gris qui est utilisé. Au total, huit images sont affichées sur l'écran, quatre déphasages verticaux et quatre horizontaux.

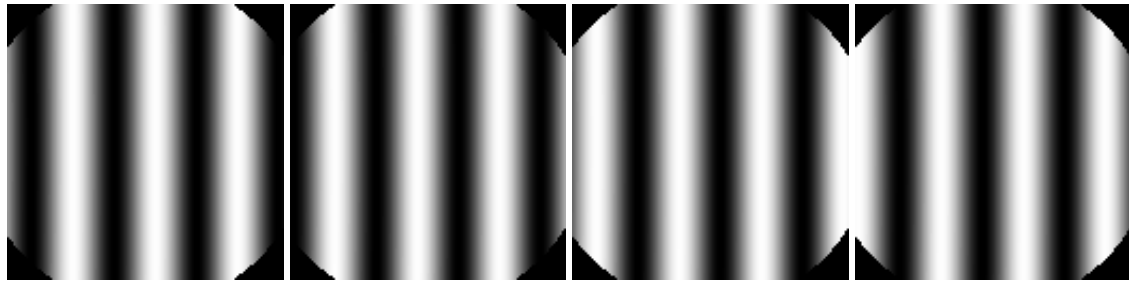


**FIGURE 3.12** – Motifs codés sur l'écran. Celui-ci affiche des franges horizontales (à gauche) et verticales (à droite). En bas : Zoom sur une période pour constater les différents niveaux de gris du motif. Les niveaux de gris sont compris dans un intervalle de  $[0, 255]$ .

Ces images sont enregistrées par la caméra et sont traitées suivant les algorithmes décrits précédemment. Le choix d'utiliser 40 périodes dans le motif à franges devient plus parlant en regardant les images réfléchies par le miroir sur la figure 3.13. En effet, le miroir ne réfléchit que trois périodes de franges. Il faut donc un motif avec suffisamment de périodes sur l'écran pour que le miroir en réfléchisse quelques-unes.

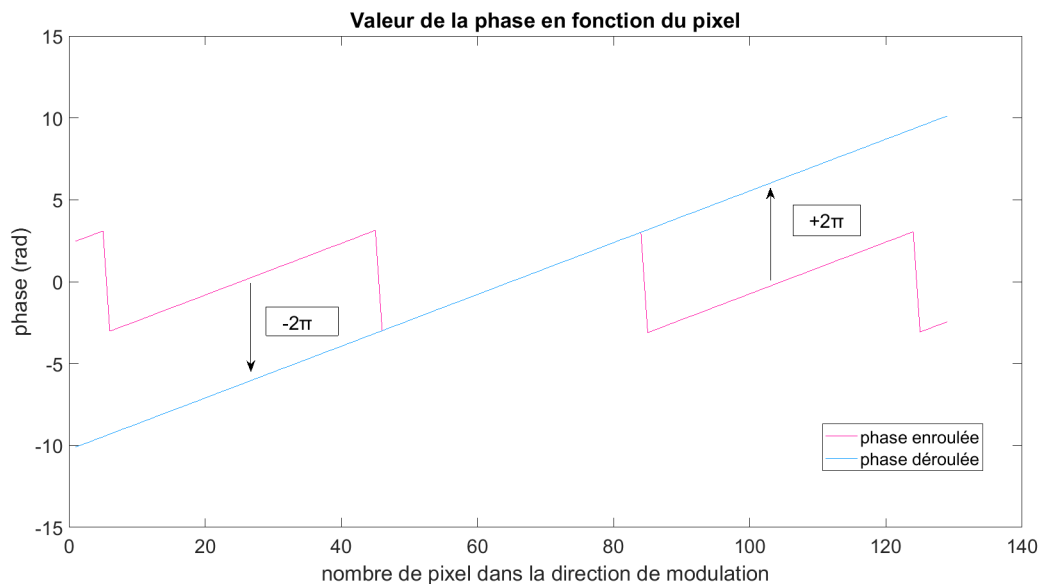


**FIGURE 3.13** – Franges réfléchies par le miroir et capturées par la caméra pour 4 déphasages horizontaux. Ces déphasages sont de gauche à droite de  $0$ ,  $\frac{\pi}{2}$ ,  $\pi$  et  $\frac{3\pi}{2}$ .

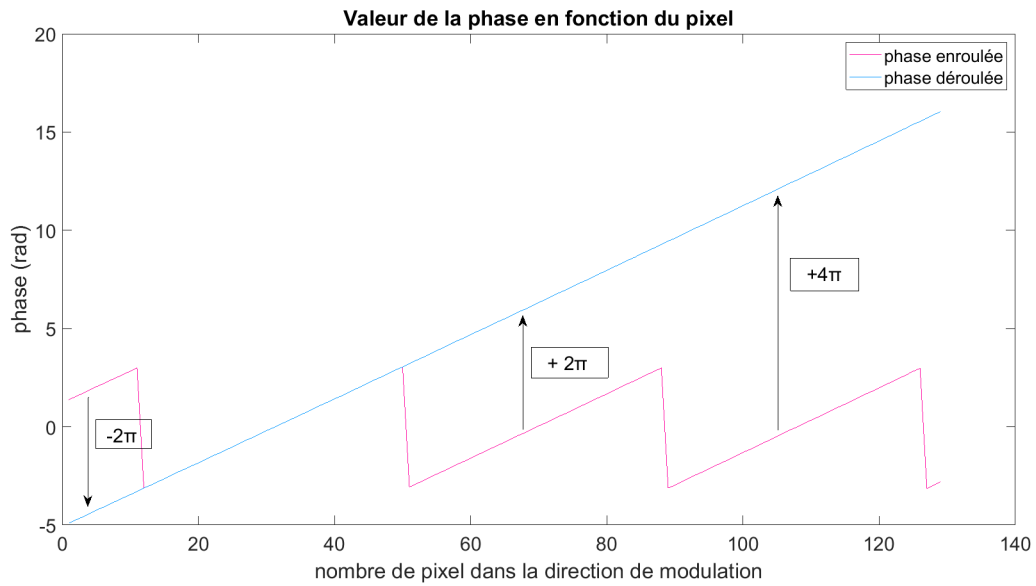


**FIGURE 3.13** – Franges réfléchies par le miroir et capturées par la caméra pour 4 déphasages verticaux. Ces déphasages sont de gauche à droite de  $0$ ,  $\frac{\pi}{2}$ ,  $\pi$  et  $\frac{3\pi}{2}$ .

L'enroulage et le déroulage de phase dans les deux directions donnent les graphiques de la figure 3.14. Les courbes sont celles attendues. En effet, la phase enroulée présente des dents de scie qui montrent les sauts de  $2\pi$  et la phase déroulée (ou phase absolue) est linéaire. Elle est obtenue en remontant ou en redescendant les portions de dents de scie (il s'agit de supprimer les sauts  $2\pi$  comme évoqué dans la théorie). Cependant, le fait que le détecteur soit de moins bonne résolution pourrait traduire un manque de précision dans le traitement de la phase. En effet, dans le cas horizontal, la phase enroulée oscille entre  $-3,1140$  et  $3,1377$  radians. Tandis que pour le cas vertical, les valeurs oscillent entre  $-3,1416$  et  $3,00669$  radians. Le nombre de franges réfléchies pourrait également jouer un rôle dans cette précision. Le déroulage de phase dans les deux directions donne bien des valeurs de phases absolues linéaires allant de  $-10,1381$  à  $10,1602$  radians pour les franges horizontales et allant de  $-4,9052$  à  $16,0470$  radians pour les franges verticales.

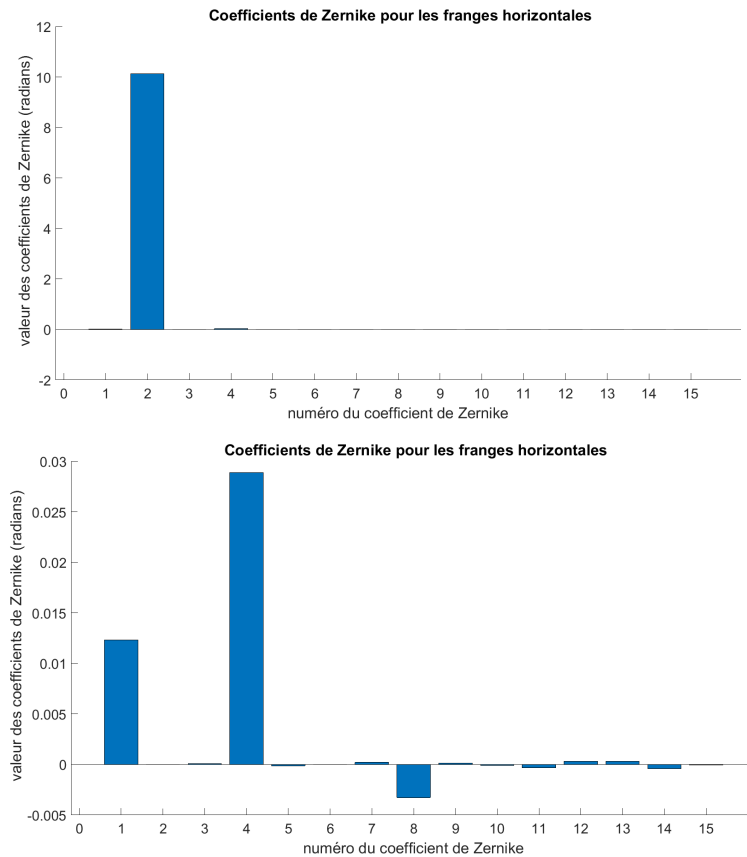


**FIGURE 3.14** – Cartes de phase enroulée et déroulée pour les franges horizontales. La phase déroulée est obtenue en supprimant les sauts de  $2\pi$  dans la structure en dents de scie.



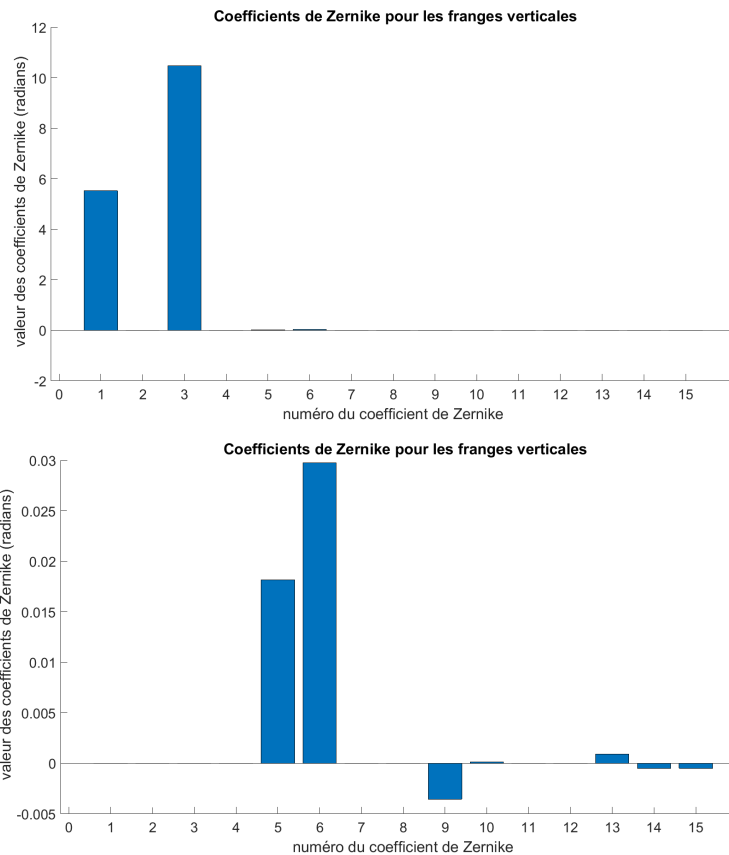
**FIGURE 3.14** – Cartes de phase enroulée et déroulée les franges verticales. La phase déroulée est obtenue en supprimant les sauts de  $2\pi$  dans la structure en dents de scie.

La carte de phase déroulée est analysée avec les polynômes de Zernike. On ne regarde que les coefficients de bas ordre car ils possèdent une grande sensibilité face à un désalignement [52]. La signature des coefficients dans le cas d'un alignement parfait est illustré à la figure 3.15. Pour ne pas encombrer les graphiques, les coefficients sont remplacés par des numéros. Ainsi, le numéro 1 est le coefficient  $C_0^0$ , le 2 =  $C_1^{-1}$ , le 3 =  $C_1^0$ , le 4 =  $C_1^1$ , le 5 =  $C_2^{-2}$ , le 6 =  $C_2^0$ , le 7 =  $C_2^2$  et ainsi de suite. Comme évoqué précédemment, seuls les premiers coefficients seront intéressants. Deux graphiques pour chaque type de franges sont réalisés. Le premier reprend toutes les valeurs des coefficients obtenus, tandis que le second est un zoom sur les valeurs plus faibles en supprimant les grandes valeurs. Cela permet d'avoir une idée plus précise sur la signature du système. L'image 3.15 montre bien que les coefficients de haut ordre ne sont pas intéressants pour notre étude d'alignement car leur présence est insignifiante face aux coefficients de bas ordre.



**FIGURE 3.15** – Coefficients de Zernike pour les franges horizontales lorsque le miroir est considéré comme aligné parfaitement. Du haut vers le bas : coefficients de Zernike pour les franges horizontales et zoom sur les coefficients plus faibles de Zernike pour les franges horizontales. Ce zoom est réalisé en supprimant les valeurs dominantes des graphiques afin de visualiser au mieux les autres valeurs de coefficients.

Il est important de noter que le comportement des 6 premiers coefficients dans le cas horizontal et vertical est opposé. En effet, pour les franges horizontales, le coefficient 2 et le coefficient 4 (dans le graphique zoom) prédominent dans les graphiques. Cependant, ils sont insignifiants dans le cas des franges verticales où ce sont les coefficients 1, 3, 5 et 6 (dans le graphique zoom pour le 5 et 6) qui prédominent. Dès lors, l'utilisation de franges dans les deux orientations enrichit l'analyse d'une configuration.



**FIGURE 3.15** – Coefficients de Zernike pour les franges verticales lorsque le miroir est considéré comme aligné parfaitement. Du haut vers le bas : coefficients de Zernike pour les franges verticales et zoom sur les coefficients plus faibles de Zernike. Ce zoom est réalisé en supprimant les valeurs dominantes des graphiques afin de visualiser au mieux les autres valeurs de coefficients.

Cette approche peut être réalisée pour tous les autres miroirs hors axe du CubeSat. A ce stade, l'idée est de vérifier si les étapes et les algorithmes choisis sont pertinents pour répondre au thème de ce mémoire.

Une étude du cas parfait a également été menée pour un motif ayant 120 périodes. Le nombre de franges perçues sur le miroir a triplé. Dès lors, les intervalles de déroulage de phase sont plus grands ayant pour conséquence des valeurs de coefficients de Zernike plus importantes. Cependant, les dominances des coefficients dans les deux directions de modulation restent les mêmes. Dans la suite de ce chapitre, le désalignement sera étudié avec la période 120.

### 3.5 Robustesse de l'algorithme de déroulage

Comme décrit précédemment, un algorithme de déroulage de phase a pour but de dérouler la phase en supprimant les sauts  $2\pi$  dans la phase enroulée. L'idée de cette section est de calculer l'erreur commise sur le déroulage de phase. Pour ce faire, la phase déroulée sera réenroulée et comparée avec la phase enroulée initiale. Un bon déroulage de phase donnera une différence nulle. L'analyse s'effectue uniquement dans un alignement parfait. De plus, le comportement de l'algorithme de dé-

roulage ainsi que du système sera analysé en injectant plusieurs types de bruits.

### 3.5.1 Pixels morts

Afin de vérifier la robustesse du code, des pixels morts de l'écran ont été simulés. Cette situation est très courante lorsqu'en laboratoire, l'expérience demande l'utilisation d'un écran. Comme le montre les images de la figure 3.16, plusieurs zones mortes de tailles variables ont été incorporées dans l'image. Ces zones restent statiques tant que l'écran ou le miroir ne bougent pas. Ces zones ne

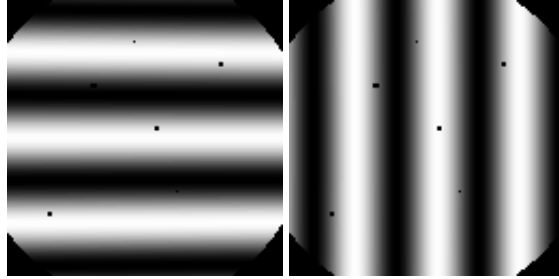


FIGURE 3.16 – Simulation de la présence de pixels morts afin de tester la robustesse de l'algorithme de déroulage.

variant pas en intensité sont donc nulle. L'enroulage de phase marque directement ces zones comme des NaN (Not a Number en anglais) car le développement divise 0 par 0 dans ces régions. Dès lors l'algorithme de déroulage ne considère automatiquement pas ces régions pour traiter la phase comme l'illustre la figure 3.17 avec les franges horizontales. L'erreur dans le déroulage de phase est très

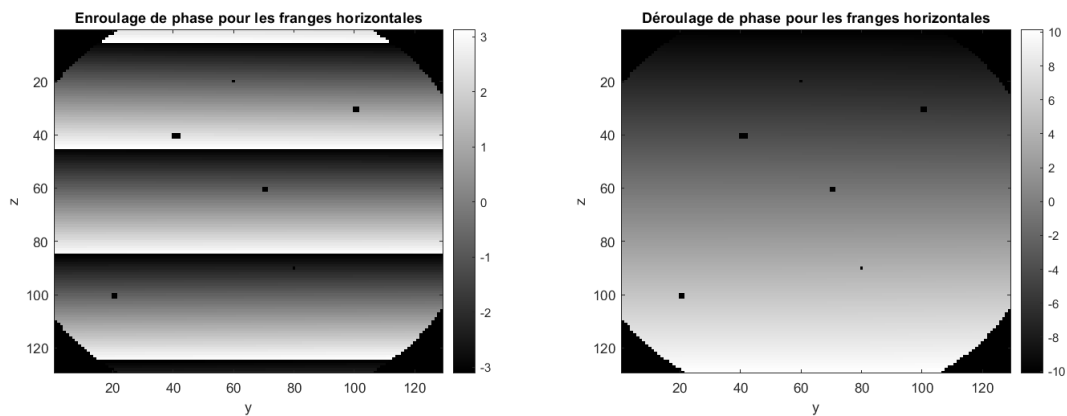
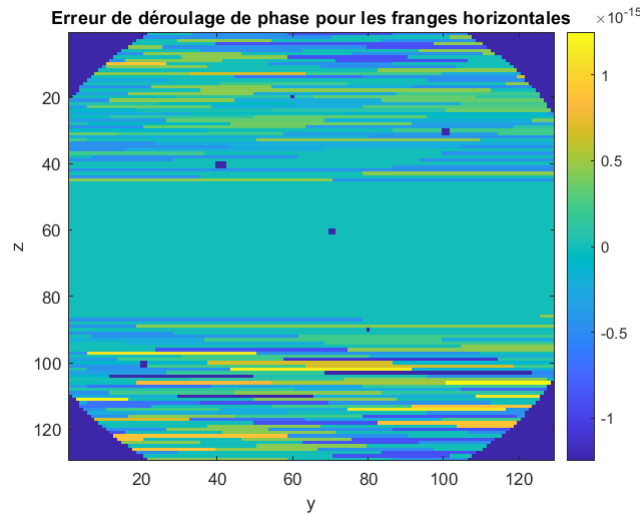


FIGURE 3.17 – Carte de phase enroulée et déroulée dans le cas de franges horizontales. La présence de pixel mort est remarqué sur ces cartes, mais ne possède pas de valeur

faible et vaut  $10^{-15}$  rad comme le montre la figure 3.18. Les bords bleu foncé sont les parties hors du miroir. Dans le traitement d'images, elles sont considérées comme étant des NaN.

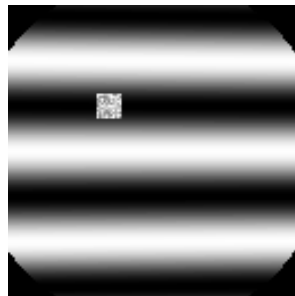
Les valeurs de déroulage de phase restent identiques que dans le cas d'une image sans zone morte.



**FIGURE 3.18** – Erreur commise sur le déroulage de phase en présence de pixels morts. Les zones en bleu foncées sont considérées comme des NaN.

### 3.5.2 Bruit aléatoire local

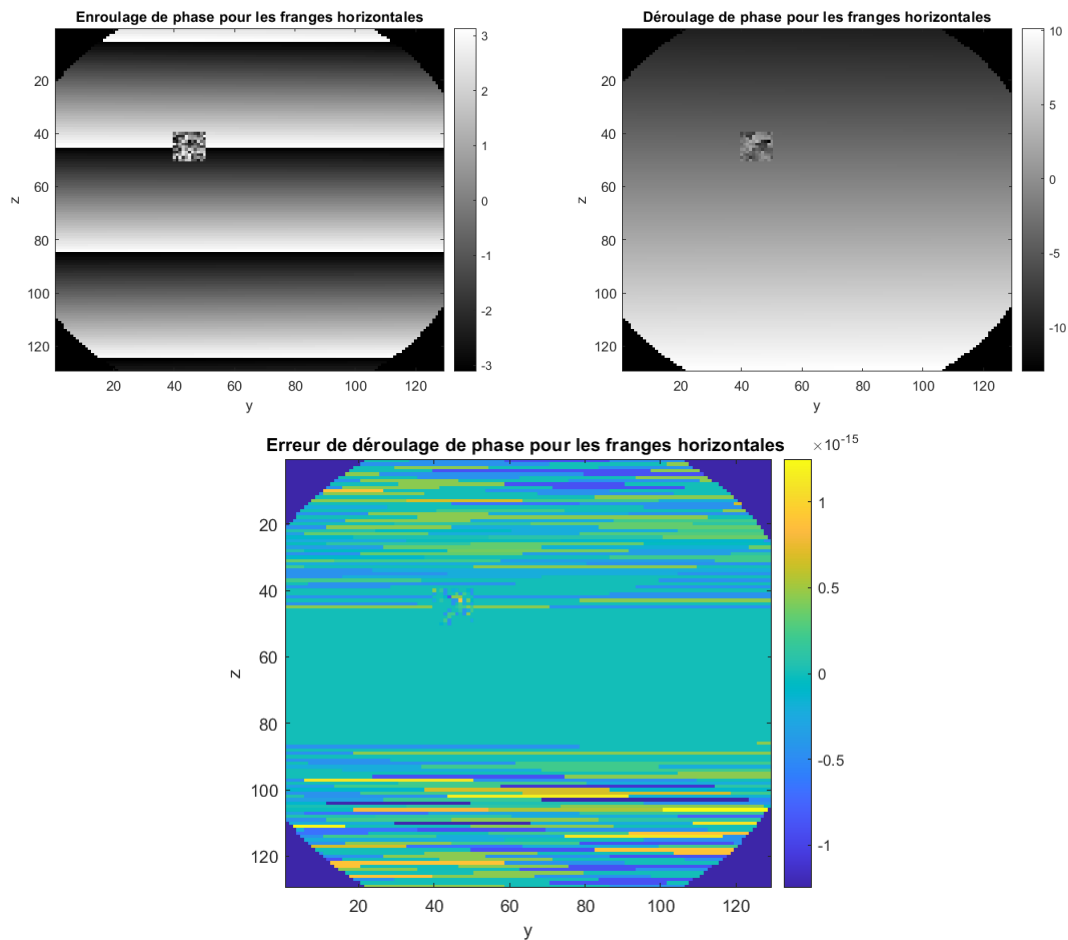
Pour cette partie, les images de franges sur le miroir sont entachées d'une petite zone de bruit aléatoire. Celle-ci est définie sur un ensemble de  $10 \times 10$  pixels comme le montre la figure 3.19. L'intérêt est de vérifier si l'algorithme déroule toujours convenablement la phase en contournant le bruit local sans le propager.



**FIGURE 3.19** – Image de franges horizontales de déphasage nulle avec une zone de bruit locale.

Après analyse des images, il est aisé de dire que l'algorithme déroule convenablement la phase avec une erreur de  $10^{-15}$  rad comme le montre la figure 3.20. De plus, le bruit local n'est pas propagé sur toute l'image. Il reste donc isolé du reste des données fiables donnant toute sa robustesse à l'algorithme de déroulage.

La différence notable avec l'injection de ce bruit est bien évidemment les valeurs de déroulage de phase, surtout pour la direction verticale où les valeurs se trouvent dans l'intervalle  $[-11, 1884, 9, 7639]$  rad. Les coefficients de Zernike s'en retrouvent également affectés non pas dans les ordres de grandeur, mais bien dans la dominance notamment pour les franges verticales. En effet, le coefficient 12 domine les coefficients 5 et 6.



**FIGURE 3.20** – En haut : phase enroulée et déroulée dans le cas de franges horizontales. Malgré la présence du bruit local, le traitement de la phase reste correcte. En bas : erreur commise sur le déroulage de phase. Celle-ci reste du même ordre de grandeur sans la présence de bruit.

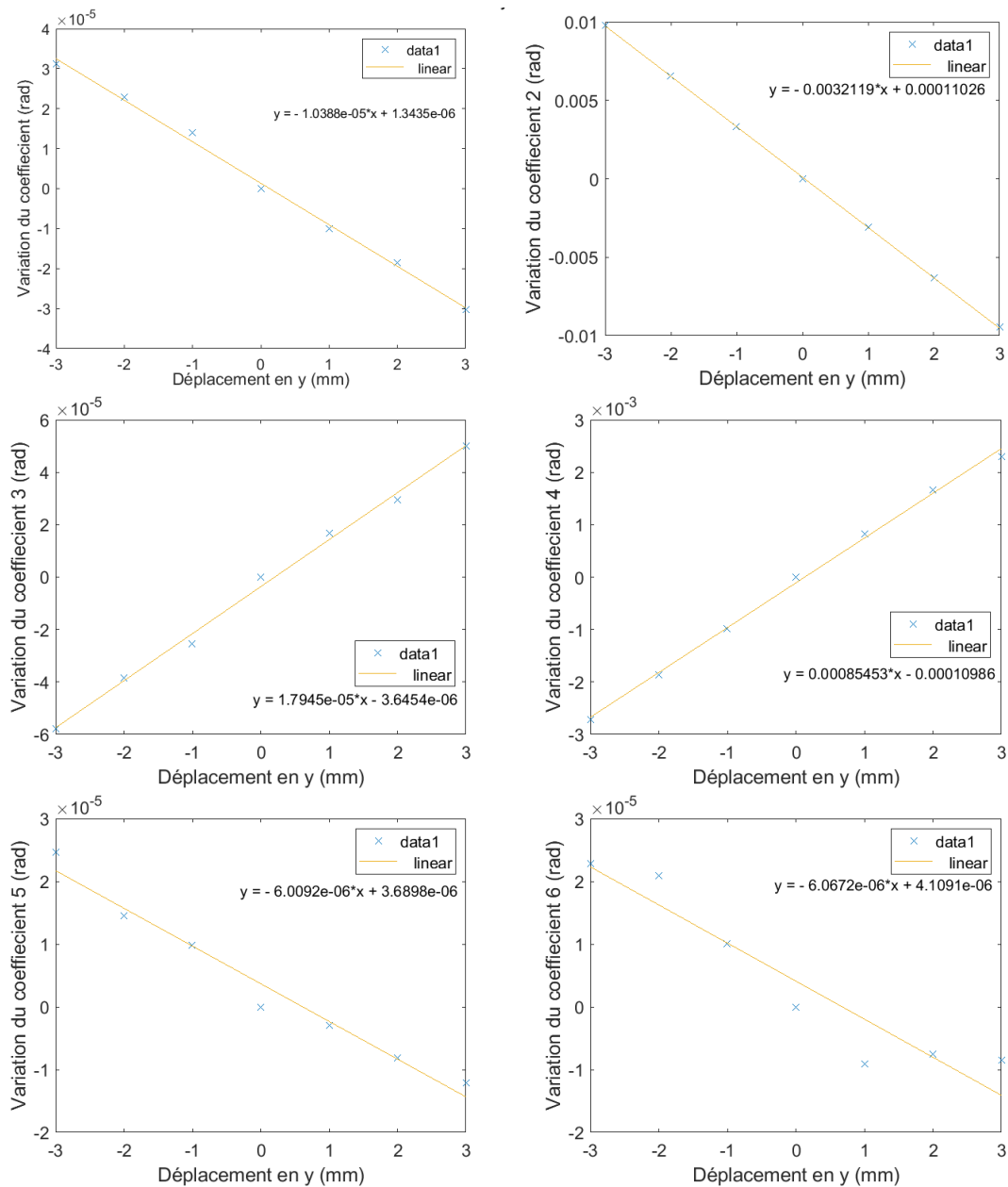
## 3.6 Désalignement individuel

### 3.6.1 Elaboration d'un modèle pour une translation en $y$

Les désalignements du miroir parabolique hors axe vont se faire individuellement, c'est-à-dire que tous les degrés de liberté vont être fixés à l'exception d'un seul. A chaque étape du désalignement, toute une analyse sur la phase sera nécessaire. En effet, le changement de position du miroir entraîne un changement dans les cartes de phase. Ce changement va être quantifié par la variation des coefficients de Zernike.

Cette section traite d'une translation en  $y$  du miroir. En effet, il est important de notifier que les désalignements doivent s'effectuer dans le système de coordonnées du miroir. Celui-ci est dans un premier temps déplacé sur  $\pm 3$  mm. A chaque étape de la translation, une analyse sur la phase est nécessaire pour en retirer les valeurs des coefficients de Zernike qui caractérisent la configuration dans laquelle se trouve le miroir. Une fois les valeurs des coefficients récupérées, il faut analyser les variations de ceux-ci par rapport au cas parfait donnant ainsi les graphiques sur la figure 3.21. Il

serait plus intéressant d'effectuer l'analyse avec plus de points de mesures, mais ce choix se justifie principalement à cause du code élaboré pour simuler le système. En effet, celui-ci n'est pas parfaitement optimisé et demande de tourner pendant plus de six heures pour fournir un point de la translation. Ceci s'explique facilement par le fait que tous les rayons du détecteur sont construits dans une double boucle for. Une des perspectives à envisager serait de passer en calcul matriciel.



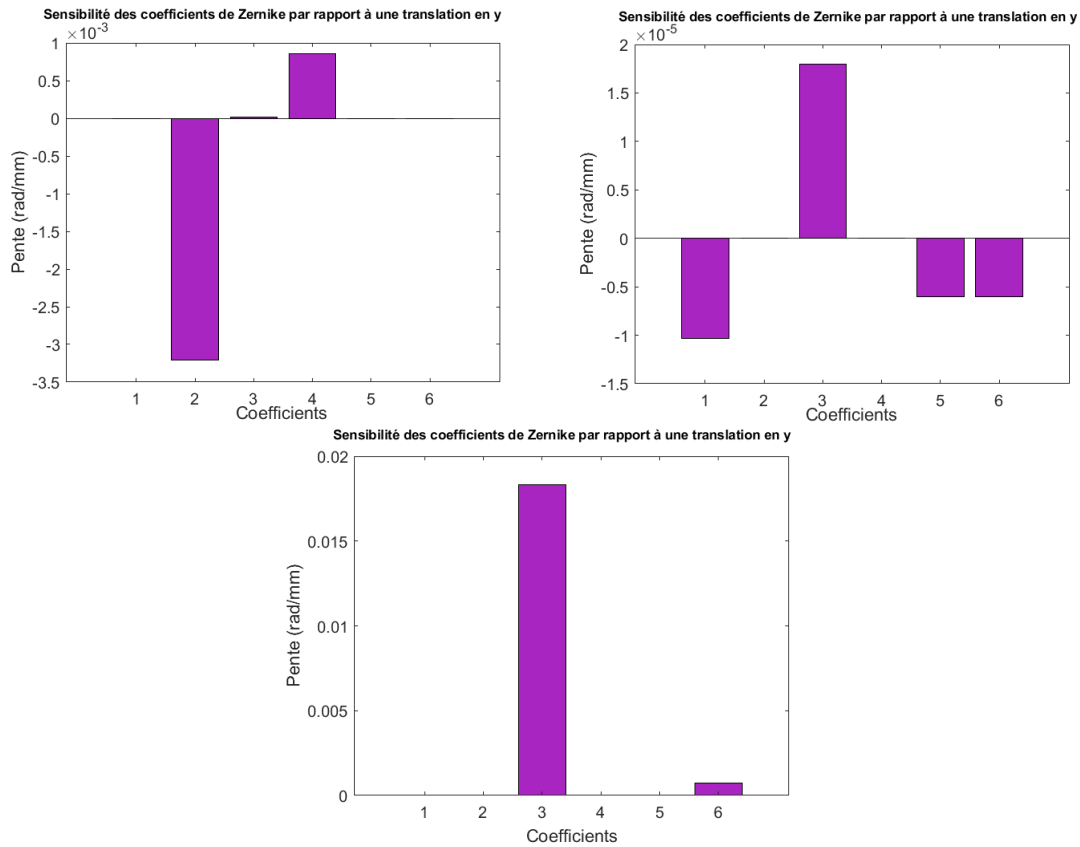
**FIGURE 3.21** – Graphiques reprenant la variation des coefficients 1, 2, 3, 4, 5 et 6 en fonction d'une translation en  $y$  de  $\pm 3$  mm dans le cas de franges horizontales. Ils présentent tous un comportement bien linéaire à l'exception du coefficient 6 où la linéarité commence à se perdre.

Les graphiques de la figure 3.21 montrent que les coefficients de bas ordre ont un comportement

linéaire devant la translation en  $y$ . Ce comportement linéaire est également retrouvé dans l'article [52]. La linéarité du comportement est validée par la fiabilité du fit linéaire. Pour cela, le coefficient de détermination  $R^2$  a été extrait pour chaque graphique. Il s'agit d'un coefficient compris entre 0 et 1 et donne le degré de fiabilité du fit (linéaire dans notre cas). Le cas horizontal possède un  $R^2$  de 0,9961 pour le coefficient 1 ; 0,9999 pour le 2 ; 0,9953 pour le 3 ; 0,9975 pour le 4 ; 0,9719 pour le 5 et 0,894 pour le 6. Plus le  $R^2$  est proche de 1 plus le fit est fiable. Dans le cas vertical,  $R^2$  vaut 0,9995 pour le coefficient 3 et 0,9443 pour le coefficient 6. La pente de chaque graphique donne la sensibilité du coefficient face à ce désalignement. Dans le cas des franges horizontales, ce comportement linéaire se retrouve pour les 6 premiers coefficients. Cependant, le coefficient 6 commence à montrer une perte de linéarité traduisant le fait qu'il soit beaucoup moins sensible que les autres coefficients. En allant plus haut dans les ordres, il est vite constaté que cette linéarité n'est plus présente. Ces coefficients seront considérés comme non sensibles. Dans le cas des franges verticales, ce comportement linéaire se retrouve pour les coefficients 3 et 6, dont les équations du fit linéaire sont respectivement  $y = 0,018325.x - 0,00027411$  et  $y = 0,00072747.x + 6,3344E - 05$ . Les autres coefficients de bas ordre ont une variation de l'ordre de  $10^{-16}$  rad/m. Leur variation est plus petite que l'erreur commise sur la phase ce qui signifie que le phénomène étudié n'intervient en rien. Ces coefficients sont donc insensibles à une translation en  $y$ .

L'étape suivante est de regrouper tous les coefficients et de déterminer les degrés de sensibilité (fortes, moyennes, faibles) de chacun. Pour ce faire, un fit linéaire est appliqué sur chaque graphique et les pentes sont récupérées pour quantifier la sensibilité. Les graphiques de la figure 3.22 permettent de conclure sur la translation en  $y$ . Les coefficients 2 et 4 dans le cas de franges horizontales ont la plus forte sensibilité. Cependant, dans le cas des franges verticales, il s'agit uniquement du coefficient 3. Cette étape dans l'analyse des données permet de procurer une signification physique aux coefficients. En effet, chaque coefficient de Zernike est associé théoriquement à un piston, tip, tilt ou des aberrations optiques lorsque les polynômes sont utilisés sur des cartes de coordonnées. Cependant, ce mémoire étudie des cartes de phase absolue. La signification des coefficients n'a plus le même sens physique dans notre contexte. Ainsi, grâce à cette analyse sur les translations en  $y$ , une nouvelle définition de chaque coefficient est établie. Une translation en  $y$  est donc caractérisée par les coefficients de plus forte sensibilité dans les deux directions de modulation. Il est intéressant de constater que les sensibilités s'opposent en fonction de la direction des franges entraînant une caractérisation plus précise de l'état de désalignement. Les informations dans les deux directions de modulation ne sont donc pas redondantes, mais bien complémentaires. Ainsi, une translation en  $y$  sera plus précisément détectée si on combine les données dans le cas horizontal et vertical. Il serait dès lors intéressant à l'avenir d'étudier la problématique en combinant directement les deux modulations étant donné qu'il vient d'être prouvé que leur utilisation n'est pas redondante.

La translation suivant  $y$  a aussi été étudiée dans le domaine des  $\mu\text{m}$ . Le déplacement du miroir s'effectue de  $-300$  à  $300 \mu\text{m}$  par pas de 100. Un premier constat est que la linéarité dans les variations des coefficients n'est pas aussi présente que dans le cas du mm. Pour les franges horizontales, un comportement linéaire est constaté pour les coefficients 1, 2 et 4, ainsi que le coefficient 3 pour les franges verticales. Une autre observation est la sensibilité de ces coefficients. En effet, autant la dominance dans les sensibilités des coefficients ne change pas (2 et 4 pour le cas horizontal et 3 pour le cas vertical) autant leur sensibilité est beaucoup plus faible d'un facteur de  $10^{-3}$ . Les relations de proportionnalité sont donc respectées.



**FIGURE 3.22** – Sensibilités des coefficients de Zernike dans le cas d’une translation en  $y$  de  $\pm 3$  mm. En haut : sensibilités des coefficients dans le cas de franges horizontales. En bas : sensibilités des coefficients dans le cas de franges verticales. Chaque modulation possède leurs propres coefficients de forte sensibilité.

### 3.6.2 Translation inconnue

Une fois que le modèle a été élaboré et que les sensibilités les plus fortes des coefficients ont été mises en lumière pour chaque direction de modulation, il serait intéressant de tester la fiabilité de celui-ci. Ainsi, le miroir est translaté en  $y$  d’une certaine distance supposée inconnue. Il faut dès lors récupérer la signature des coefficients de Zernike en effectuant une analyse sur les cartes de phase comme précédemment. Les coefficients récupérés sont ajustés avec l’offset établi dans le cas parfait, c’est-à-dire qu’on détermine l’écart en radian que chaque coefficient de Zernike par rapport au coefficient de l’alignement parfait. En se basant sur les graphiques de sensibilité des coefficients dans la figure 3.22, l’étude d’une translation en  $y$  se focalise sur les écart des coefficients 2 et 4 pour la modulation horizontale et sur l’écart du coefficient 3 pour la modulation verticale. Ces valeurs sont injectées dans l’équation du fit linéaire de la figure 3.21 correspondant à leur numéro. Trois résultats sont obtenus : 1,8162 mm pour le coefficient 2, 1,5568 mm pour le coefficient 4 et 1,6827 mm pour le coefficient 3. Ces valeurs ensuite sont moyennées. Au final, une translation en  $y$  de 1,6853 mm a été détectée. En réalité, l’algorithme a simulé une translation de 1,7 mm. Le modèle peut donc être validé.

# Chapitre 4

## Perspectives et conclusion

### 4.1 Perspectives

Cette section propose différents chemins pour continuer à bien ce projet. En effet, ce mémoire s'arrête avant d'apporter une conclusion franche quant à la possibilité de placer un système de défectométrie sur un CubeSat. Beaucoup d'horizons restent à explorer dont une synthèse est faite ci-dessous.

#### 4.1.1 Algorithmes

L'algorithme de ray tracing doit absolument être optimisé afin de réduire les temps de calcul. De plus, la configuration étudiée n'est pas physiquement réaliste. En effet, l'écran se trouve devant la caméra et obstrue donc son champ de vue. Dès lors, il serait préférable à l'avenir de travailler dans une situation se rapprochant de la réalité. Il est également envisageable d'utiliser directement un logiciel de modélisation optique comme Zemax afin d'augmenter considérablement le réalisme dans les simulations. Cependant, la prise en main n'est pas évidente.

L'étude du sujet de ce mémoire pourrait aussi prendre en compte une comparaison entre les algorithmes de déroulage de phase et de choisir celui qui répond le mieux aux exigences de laboratoire ou utiliser des masques pour cacher les zones bruitées et voir l'influence sur les coefficients de Zernike. Etant donné que l'analyse de la problématique n'est qu'à ses débuts, s'il s'avère qu'un fit de Zernike ne soit pas concluant pour les autres types de déplacements, il existe toutes sortes de polynômes comme Chebychev, B-spline, etc. En effet, les polynômes de Zernike ne sont peut-être pas les mieux adaptés pour analyser la sensibilité des coefficients.

#### 4.1.2 Traitement des données

Dans un premier temps, cette étude doit continuer avec un nombre de mesures plus important afin de rendre l'analyse plus pertinente. De plus, une étude plus approfondie notamment sur la période optimale pourrait être intéressante pour générer le meilleur motif possible.

Comme évoquée dans le chapitre précédent, l'idée de l'analyse des données suivra l'article [52]. La suite logique de ce travail est de continuer tout d'abord l'analyse des désalignements individuellement, à savoir les translations selon  $x$  et en  $z$ , mais aussi les rotations autour des axes  $x$ ,  $y$  et  $z$ . L'enjeu de cette partie du travail est de voir si les coefficients de Zernike sont capables de faire la distinction entre un déplacement de franges due à une translation ou une rotation.

Etant donné que l'objectif final est de connaître la position inconnue du miroir sur base des cartes de phase, il est nécessaire d'étudier aussi les combinaisons de désalignements. En effet, une fois déployé, les miroirs hors axe peuvent bouger suivant les 6 degrés de liberté en même temps. L'étude du comportement des coefficients de Zernike dans le cas de mouvements combinés est primordiale afin de caractériser la signature des coefficients sur le désalignement inconnu. Pour ce faire, un modèle mathématique doit être construit reliant la sensibilité de chaque coefficient aux désalignements étudiés (indépendants ou combinés).

Toute cette analyse doit porter sur tous les miroirs du CubeSat. Ainsi, quatre modèles linéaires seraient obtenus.

### 4.1.3 Mise en oeuvre

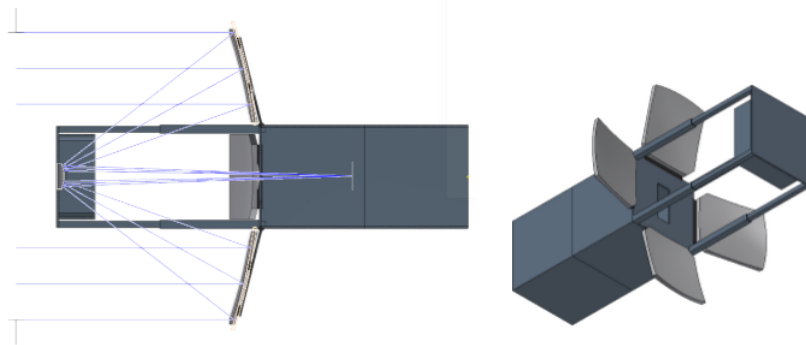
Il serait intéressant de pousser la calibration du système au-delà de la correction de la non-linéarité de l'écran et de constater si la précision dans les mesures respecte bien celle souhaitée pour la problématique étudiée, c'est-à-dire l'ordre de grandeur du micromètre. Il se pourrait que certaines calibrations ne soient dès lors pas nécessaires dans notre cas.

Dans un premier temps, le dispositif imaginé dans le chapitre 2 de ce mémoire n'est faisable que pour des tests au sol. En effet, l'idée est de mesurer les miroirs primaires et de déterminer la limite de résolution de la technique. Pour ce faire, le dispositif est placé devant les miroirs sans se soucier de l'obstruction du champ de vue. La position du dispositif de métrologie n'est donc pas une priorité dans ce cas. Seule la robustesse du montage est testée. Qui plus est, cette partie du travail pourrait juste se focaliser sur la précision du déploiement sur des barres de flexion. Ainsi, une fois que le niveau de confiance sur la mécanique de déploiement est acquis, le projet peut uniquement rester au stade de tests au sol.

Cependant, si l'idée finale est de placer le dispositif sur un CubeSat en vol, la compacité et l'emplacement des instruments de mesure deviennent des enjeux majeurs afin d'avoir la possibilité de vérifier la position des miroirs dans l'espace. Dès lors, tout le montage doit être miniaturisé que ce soit au niveau de la caméra qu'au niveau de l'écran. Plusieurs idées de configurations ont déjà été envisagées. Par exemple, la figure 4.1 pourrait servir de base pour élaborer un dispositif de déflectométrie dans l'espace. Une des idées qui pourrait être étudiée pour la suite de ce sujet de mémoire serait de considérer le détecteur à l'intérieur du CubeSat comme étant le détecteur de la caméra et imaginer des petits écrans déployables dans la coiffe des miroirs secondaires. Cependant, le nombre d'écrans serait égal aux nombres de miroirs augmentant la charge utile du CubeSat. Une autre idée serait de placer un écran et une caméra à la place des miroirs secondaires durant le temps de réglage afin de mesurer tous les miroirs en même temps. De plus, il faudrait imaginer que le déploiement ne soit pas totalement fini pour éviter toute interférence avec les rayons issus de l'infini comme le montre l'image de gauche sur la figure 4.1. Dès lors, un bord de chaque miroir pourrait être biseauté afin que les rayons issus de l'écran soient renvoyés vers la caméra. Le désavantage est la diminution de la résolution car ce petit bord ne participera pas à l'imagerie du CubeSat.

## 4.2 Conclusions

Ce mémoire se propose d'ouvrir les portes sur une application de la déflectométrie sur un CubeSat. L'objectif n'est pas d'apporter une réponse franche et directe telle que oui ou non sur la possibilité d'une mise en oeuvre, mais plutôt de soumettre une sorte de guide de démarrage dans le domaine et d'envisager l'application d'une méthode.



**FIGURE 4.1** – Schéma d'un CubeSat avec sa structure de déploiement. Sa conception peut servir de base afin d'imaginer la mise en oeuvre d'un système de déflectométrie à bord d'un CubeSat. [70]

Cette étude ne peut commencer sans prendre connaissance du contexte dans lequel cette technique doit être mise en oeuvre. En effet, le domaine d'application va influencer l'élaboration d'un système de déflectométrie. Cette analyse préalable conditionne les recherches et le choix du système à mettre en place. Suite aux différentes recherches menées durant une partie de l'année, une technique a été élaborée pour tenter de répondre à la question de faisabilité. Elle est basée sur une configuration de type Hartmann inversé, c'est-à-dire une configuration la plus simple possible et avec le minimum d'éléments (écran, caméra et miroir parabolique hors axe). Les mesures sont basées sur les déphasages d'une fonction cosinus de modulation horizontale et verticale. Ainsi, le miroir peut être caractérisé par des cartes de phase. L'état d'alignement du miroir est déterminé par la signature des coefficients de Zernike, dont un fit a été réalisé sur ces cartes de phase.

La mise en oeuvre d'une technique rencontre également des erreurs systématiques dont certaines, comme la non-linéarité dans l'affichage des intensités de l'écran ont pu être corrigées. Après calibration du système, la non-linéarité de l'écran est contrebalancée avec une erreur maximale de  $\pm 1$  en intensité de niveaux de gris.

Malheureusement, la crise pandémique du COVID-19 a engendré la fermeture de tous les laboratoires, empêchant la continuation de cette étude d'un point de vue expérimentale. Dès lors, ce mémoire a dû être réorienté dans la façon d'aborder le problème, ce qui explique le passage intensif aux simulations. La méthodologie établie pour l'expérimentation a donc été reprise sous forme numérique. La fiabilité de tous les codes a été testée, ainsi que la robustesse de ceux-ci. L'erreur commise sur la position d'un rayon est de l'ordre de  $10^{-6}$  m ce qui conditionne l'ordre de grandeur des déplacements étudiés. Par manque de temps, seul un modèle pour une translation en  $y$  a pu être analysé dans le domaine des millimètres, mais aussi dans les centaines de micromètres. Un modèle a été établi pour les translations en  $y$  et a été testé pour un désalignement inconnu selon ce degré de liberté. Suite à l'analyse de cette configuration inconnue, le modèle estime le déplacement du miroir avec une erreur de 0.01 mm. Il est donc pertinent et mérite d'être approfondi. Cependant, un grand défi évoqué dans les perspectives est de pouvoir caractériser et différencier chaque degré de liberté, à savoir 3 translations et 3 rotations.

Ce mémoire peut se conclure de manière générale en disant que la technique élaborée n'est pas à proscrire dans un premier temps. Elle fournit des observations intéressantes qui méritent d'être approfondies. Actuellement, la méthode ne répond pas encore aux exigences du contexte, mais elle doit retenir notre attention afin d'explorer ses limites. En effet, il reste énormément de défis pour réussir à trouver les désalignements sur les 6 degrés de liberté, surtout à la précision requise pour

une application sur CubeSat. L'aspect technique tel que la miniaturisation doit encore être pris en compte.

# Bibliographie

- [1] Tom Mladenov. Fpga-based active pointing correction of optical instruments on small satellites. Master's thesis, UHasselt, 2018.
- [2] Rêves d'espace : Cubesat, c'est quoi? ça sert à quoi? [https : //reves - d - espace.com/cubesat - c - est - quoi - ca - sert - a - quoi/](https://reves-d-espace.com/cubesat-c-est-quoi-ca-sert-a-quoi/). Accessed : 2019-10-19.
- [3] Jérôme Wertz, Vincent Lenaerts, Olivier Bruls, and Amandine Denis. Conception et réalisation du système de déploiement des antennes du nanosatellite oufti-1. *Revue Scientifique des Instituts Supérieurs Industriels Libres Francophones, volume 24*, pages 1–25, 2010.
- [4] Martin Sweeting. Modern small satellites-changing the economics of space. *Proceedings of the IEEE*, 106(3) :343–361, 2018.
- [5] Sébastien Duijsens. *Preliminary study of a deployable CubeSat*. PhD thesis, Université de Liège, 2016-2017.
- [6] S Lee, A Hutputanasin, A Toorian, W Lan, R Munakata, J Carnahan, D Pignatelli, and Arash Mehrparvar. Cubesat design specification, rev. 13 : The cubesat program. *California Polytechnic State Univ., San Luis Obispo*, 2014.
- [7] Université de liège : Oufi-1. [http : //events.ulg.ac.be/oufti - 1/oufti - 1/](http://events.ulg.ac.be/oufti-1/oufti-1/). Accessed : 2020-03-01.
- [8] Club cubesat : conception de nano-satellites à l'isae-superaero. [https : //cubesat.bde - supraero.fr/eye - sat/](https://cubesat.bde-superaero.fr/eye-sat/). Accessed : 2019-12-19.
- [9] Groupe isae : Lancement réussi pour eyesat, observateur de poche de la voie lactée. [https : //www.isae - supraero.fr/fr/actualites/lancement - reussi - pour - eyesat - observateur - de - poche - de - la - voie - lactee/](https://www.isae-superaero.fr/fr/actualites/lancement-reussi-pour-eyesat-observateur-de-poche-de-la-voie-lactee/). Accessed : 2020-04-20.
- [10] François Guinot. *Déploiement régulé de structures spatiales : vers un modèle unidimensionnel de mètre-ruban composite*. PhD thesis, 2011.
- [11] Pernelle Marone-Hitz. *Modélisation de structures spatiales déployées par des mètres ruban : vers un outil métier basé sur des modèles de poutre à section flexible et la méthode asymptotique numérique*. PhD thesis, 2014.
- [12] Michael E Dearborn, Geoff P Andersen, Olha Asmolova, Richard L Balthazor, Matthew G McHarg, Heather C Nelson, Trey S Quiller, Gabe R Wilson, Thomas J Harvey, and Thomas W Murphey. A deployable membrane telescope payload for cubesats. *JoSS*, 3(1) :253–264, 2014.
- [13] Pau Mallol Parera. *Deployment simulations of a composite boom for small satellites*. PhD thesis, KTH Royal Institute of Technology, 2013.
- [14] Giulia E Fenci and Neil GR Currie. Deployable structures classification : A review. *International Journal of Space Structures*, 32(2) :112–130, 2017.

- [15] Noah Schwartz, David Pearson, Stephen Todd, Maria Milanova, William Brzozowski, Andy Vick, David Lunney, Donald MacLeod, Steve Greenland, Jean-François Sauvage, et al. Laboratory demonstration of an active optics system for high-resolution deployable cubesat. *arXiv preprint arXiv:1809.09097*, 2018.
- [16] Zernike polynomials. <http://wyant.optics.arizona.edu/zernikes/zernikes.htm>. Accessed : 2020-03-08.
- [17] Noah Schwartz, David Pearson, Stephen Todd, Andy Vick, David Lunney, and Donald MacLeod. A segmented deployable primary mirror for earth observation from a cubesat platform. In *Small Satellite Conference*.
- [18] Yves Surrel. Les techniques optiques de mesure de champ : essai de classification. *Instrumentation, Mesure, Métrologie*, 4(3-4) :11–42, 2004.
- [19] Christian Faber, Evelyn Olesch, Roman Krobot, and Gerd Häusler. Deflectometry challenges interferometry : the competition gets tougher ! In *Interferometry XVI : Techniques and Analysis*, volume 8493, page 84930R. International Society for Optics and Photonics, 2012.
- [20] Markus C Knauer, Jurgen Kaminski, and Gerd Hausler. Phase measuring deflectometry : a new approach to measure specular free-form surfaces. In *Optical Metrology in Production Engineering*, volume 5457, pages 366–376. International Society for Optics and Photonics, 2004.
- [21] Guillaume Butel. *Analysis and New Developments Towards Reliable and Portable Measurements in Deflectometry*. The University of Arizona., 2013.
- [22] KK Szwedowicz. *3D-deflectometry : fast nanotopography measurement for the semiconductor industry*. Technische Universiteit Eindhoven, 2006.
- [23] Holger H Rapp. *Reconstruction of specular reflective surfaces using auto-calibrating deflectometry*, volume 23. KIT Scientific Publishing, 2012.
- [24] R Lampalzer. *Physikalische und informationstheoretische Eigenschaften und Grenzen von Systemen zur optischen Formerfassung und Inspektion nach dem Prinzip der phasenmessenden Triangulation*. PhD thesis, PhD thesis, University of Erlangen-Nuremberg, 2003.
- [25] Peng Su, Robert E Parks, Lirong Wang, Roger P Angel, and James H Burge. Software configurable optical test system : a computerized reverse hartmann test. *Applied Optics*, 49(23) :4404–4412, 2010.
- [26] Thomas Bolen et al. Élaboration d’un système de mesure de surface non-réfléchissante dans le visible par déflectométrie infrarouge. *Université de Liège, Liège, Belgique*, 2019.
- [27] Run Huang, Peng Su, Todd Horne, Guido Brusa, and James H Burge. Optical metrology of a large deformable aspherical mirror using software configurable optical test system. *Optical Engineering*, 53(8) :085106, 2014.
- [28] Lirong Wang, Peng Su, Robert E Parks, Jose M Sasian, and James H Burge. A low-cost, flexible, high dynamic range test for free-form illumination optics. In *International Optical Design Conference*, page ITuE3. Optical Society of America, 2010.
- [29] Peng Su, Manal Khreishi, Tianquan Su, Run Huang, Margaret Z Dominguez, Alejandro V Maldonado, Guillaume P Butel, Yuhao Wang, Robert E Parks, and James H Burge. Aspheric and freeform surfaces metrology with software configurable optical test system : a computerized reverse hartmann test. *Optical Engineering*, 53(3) :031305, 2013.
- [30] Zonghua Zhang, Yuemin Wang, Shujun Huang, Yue Liu, Caixia Chang, Feng Gao, and Xiangqian Jiang. Three-dimensional shape measurements of specular objects using phase-measuring deflectometry. *Sensors*, 17(12) :2835, 2017.

- [31] Lei Huang, Mourad Idir, Chao Zuo, and Anand Asundi. Review of phase measuring deflectometry. *Optics and Lasers in Engineering*, 107 :247–257, 2018.
- [32] Ping Zhao, Nan Gao, Zonghua Zhang, Feng Gao, and Xiangqian Jiang. Performance analysis and evaluation of direct phase measuring deflectometry. *Optics and lasers in engineering*, 103 :24–33, 2018.
- [33] Yue Liu, Shujun Huang, Zonghua Zhang, Nan Gao, Feng Gao, and Xiangqian Jiang. Full-field 3d shape measurement of discontinuous specular objects by direct phase measuring deflectometry. *Scientific reports*, 7(1) :10293, 2017.
- [34] Xiaoting Deng, Nan Gao, and Zonghua Zhang. A calibration method for system parameters in direct phase measuring deflectometry. *Applied Sciences*, 9(7) :1444, 2019.
- [35] Yongjia Xu, Feng Gao, and Xiangqian Jiang. Enhancement of measurement accuracy of optical stereo deflectometry based on imaging model analysis. *Optics and Lasers in Engineering*, 111 :1–7, 2018.
- [36] Yongjia Xu, Feng Gao, and Xiangqian Jiang. Performance analysis and evaluation of geometric parameters in stereo deflectometry. *Engineering*, 4(6) :806–815, 2018.
- [37] Lei Huang, Junpeng Xue, Bo Gao, and Mourad Idir. Three-dimensional shape measurement with modal phase measuring deflectometry. In *Fifth International Conference on Optical and Photonics Engineering*, volume 10449, page 1044909. International Society for Optics and Photonics, 2017.
- [38] Yuxiang Wu, Huimin Yue, Jingya Yi, Mingyang Li, and Yong Liu. Phase error analysis and reduction in phase measuring deflectometry. *Optical Engineering*, 54(6) :064103, 2015.
- [39] Run Huang. High precision optical surface metrology using deflectometry. 2015.
- [40] Arnaud Frich. Guide de la gestion des couleurs : le gamma. <https://www.guide-gestion-des-couleurs.com/gamma-ecran.html>. Accessed : 2020-02-14.
- [41] Présentation technique niveau de précision gamma 2.2. [https://www.viewsonic.com/fr/products/lcd/pdf/ti\\_gamma\\_fr.pdf](https://www.viewsonic.com/fr/products/lcd/pdf/ti_gamma_fr.pdf). Accessed : 2020-02-14.
- [42] Jianhua Wang, Fanhuai Shi, Jing Zhang, and Yuncai Liu. A new calibration model of camera lens distortion. *Pattern recognition*, 41(2) :607–615, 2008.
- [43] Juyang Weng, Paul Cohen, and Marc Herniou. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (10) :965–980, 1992.
- [44] Camera calibration toolbox for matlab. <http://www.vision.caltech.edu/bouguetj/calib-doc/htmls/example.html>. Accessed : 2020-02-12.
- [45] Edmund optics. <https://www.edmundoptics.com/p/508-x-3810mm-pfl-15-off-axis-parabolic-aluminum-mirror/33430/>. Accessed : 2020-02-01.
- [46] Edmund optics : Off-axis parabolic mirror selection guide. <https://www.edmundoptics.com/knowledge-center/application-notes/optics/off-axis-parabolic-mirror-selection-guide/>. Accessed : 2020-01-22.
- [47] Edmund optics. <https://www.edmundoptics.eu/p/50508mm-diameter-kinematic-mount-3-screws/15652/>. Accessed : 2020-01-22.
- [48] Thorlabs. [https://www.thorlabs.com/newgrouppage9.cfm?objectgroup\\_id=4024](https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=4024). Accessed : 2020-02-01.

- [49] Thorlabs : Thorcam™ software for scientific and compact usb cameras. [https : //www.thorlabs.com/software\\_pages/ViewSoftwarePage.cfm?Code = ThorCam](https://www.thorlabs.com/software_pages/ViewSoftwarePage.cfm?Code = ThorCam). Accessed : 2020-02-10.
- [50] Youtube : Thorcam software quick start. [https : //www.youtube.com/watch?v = EA\\_tLC0fg5I](https://www.youtube.com/watch?v = EA_tLC0fg5I). Accessed : 2020-02-11.
- [51] Swan Yvik. *Cours de statistique des données expérimentales de la physique*. Université de Liège, 2016-2017.
- [52] Angela Davies, Trent Vann, Christopher Evans, and Mark Butkiewicz. Phase measuring deflectometry for determining 5 dof misalignment of segmented mirrors. In *Applied Optical Metrology II*, volume 10373, page 103730H. International Society for Optics and Photonics, 2017.
- [53] Wolfram mathworld : Parabola. [https : //mathworld.wolfram.com/Parabola.html](https://mathworld.wolfram.com/Parabola.html). Accessed : 2020-04-08.
- [54] Sébastien Morterolle. *Etude de structures légères déployables pour applications spatiales*. PhD thesis, 2011.
- [55] Martin John. *Cours d'électromagnétisme*. Université de Liège, 2015-2016.
- [56] Mathonet Pierre. *Cours de géométrie*. Université de Liège, 2014-2015.
- [57] Serge Belongie et al. Rodrigues' rotation formula. *From MathWorld—A Wolfram Web Resource, created by Eric W. Weisstein*. [http : //mathworld.wolfram.com/RodriguesRotationFormula.html](http://mathworld.wolfram.com/RodriguesRotationFormula.html), 1999.
- [58] Eugene Hecht. *Optique*. Pearson education, 2005.
- [59] Anne Beyaert-Geslin. Crénelage, capiton et métadiscours (où l'image numérique résiste à la ressemblance). *Protée*, 32(2) :75–83, 2004.
- [60] Michel Desvignes, Serge Langlois, Jean-Marc Constans, and Marinette Revenu. Déroulement de phase : application à la correction de distorsions géométriques en irm. 2000.
- [61] Miguel Arevallilo Herráez, David R Burton, Michael J Lalor, and Munther A Gdeisat. Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a non-continuous path. *Applied optics*, 41(35) :7437–7444, 2002.
- [62] Mathworks : Fast 2d phase unwrapping. [https : //fr.mathworks.com/matlabcentral/fileexchange/65565 - fast - 2d - phase - unwrapping](https://fr.mathworks.com/matlabcentral/fileexchange/65565-fast-2d-phase-unwrapping). Accessed : 2020-03-04.
- [63] Miguel Arevalillo-Herráez, Francisco R Villatoro, and Munther A Gdeisat. A robust and simple measure for quality-guided 2d phase unwrapping algorithms. *IEEE Transactions on Image Processing*, 25(6) :2601–2609, 2016.
- [64] Yves Surrel. *Optique c1-18851 images optiques; mesures 2d et 3d* photocopié de cours. *Conservatoire National des Arts et Métiers*, 2003.
- [65] Eric P Goodwin and James C Wyant. *Field guide to interferometric optical testing*. SPIE Bellingham, WA, 2006.
- [66] Vasudevan Lakshminarayanan and Andre Fleck. Zernike polynomials : a guide. *Journal of Modern Optics*, 58(7) :545–561, 2011.
- [67] Mathworks, technical articles and newsletters : Analyzing lasik optical data using zernike functions. [https : //nl.mathworks.com/company/newsletters/articles/analyzing - lasik - optical - data - using - zernike - functions.html](https://nl.mathworks.com/company/newsletters/articles/analyzing-lasik-optical-data-using-zernike-functions.html). Accessed : 2020-02-14.

- [68] Wikipedia : Zernikes polynomials. *https : //en.wikipedia.org/wiki/Zernike – polynomials*. Accessed : 2020-01-22.
- [69] Mathworks : Zernike polynomials. *https : //fr.mathworks.com/matlabcentral/fileexchange/7687–zernike – polynomials*. Accessed : 2020-03-04.
- [70] Junquan Li, Steve Greenland, Craig Clark, Mark Post, Andy Vick, David Pearson, David Lee, and Donald MacLeod. Practical strategies to stabilize a nanosatellite platform with a space camera and integrated mechanical parts. In *65th International Astronautical Congress (IAC 2014)*, 2014.