

# UNIVERSITÉ DE LIMOGES

ÉCOLE DOCTORALE Sciences et Ingénierie pour l'Information,  
Mathématiques

FACULTÉ DES SCIENCES ET TECHNIQUES

## Thèse

pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE LIMOGES

Discipline : Synthèse d'Images Réalistes

présentée et soutenue par

**Reda CHAKIB**

le 14 décembre 2018

**Acquisition et rendu 3D réaliste à partir  
de périphériques « grand public »**

Thèse dirigée par Stéphane MERILLOU, Pierre-Jean VINCENT et Nicolas  
MERILLOU

### JURY :

<b>Jean-Pierre JESSEL</b>	Professeur, Université Paul Sabatier de Toulouse	Président
<b>Xavier PUEYO</b>	Professeur, Université de Gérone	Rapporteur
<b>Vincent JOLIVET</b>	Maitre de conférences, Université de Limoges	Examineur
<b>Pierre-Jean VINCENT</b>	Dirigeant de CORUO, CORUO	Examineur
<b>Stéphane MERILLOU</b>	Professeur, Université de Limoges	Examineur



## *Remerciements*

Je tiens à remercier vivement mon encadrant en entreprise M. Pierre-Jean VINCENT pour son encadrement et son soutien permanent ainsi que sa patience tout au long de cette thèse.

Je tiens aussi à remercier chaleureusement mes deux encadrants académiques M. Stéphane MERILLOU et M. Nicolas MERILLOU pour leurs assistance et leurs précieux conseils qui m'ont permis d'arriver au bout de ce travail.

Mes remerciements vont aux membres du jury qui ont pris le soin de lire et d'évaluer ce travail.

J'aimerais aussi remercier mes anciens et actuels collègues à la société CORUO avec qui j'ai eu et je partage toujours d'agréables moments et plus particulièrement Mlle. Loubna ZEALOUK avec qui nous avons collaborée sur la partie BRDF/rendu de cette thèse dans le cadre de son stage de fin d'études M2 ISICG encadré en partie par moi-même.

Je remercie aussi tous mes professeurs et anciens collègues du master ISICG.

Au final, je ne remercierai jamais assez mes parents, ma sœur et mon frère pour leurs soutien et encouragement inconditionnel durant ce travail.



# Table des matières

<b>Table des matières</b>	<b>3</b>
<b>Table des figures</b>	<b>7</b>
<b>Liste des tableaux</b>	<b>10</b>
<b>Introduction générale</b>	<b>13</b>
1 Contexte	14
2 Objectif	15
2.1 Acquisition	15
2.2 Rendu	15
3 Contributions	15
4 Organisation de la thèse	16
<b>I État de l’art</b>	<b>17</b>
1 Acquisition de la géométrie	18
1.1 Techniques d’acquisition de type optique sans-contact	18
1.1.1 Le modèle sténopé	18
1.1.2 Les paramètres intrinsèques et extrinsèques	20
1.1.3 La triangulation stéréo passive	21
1.1.4 Méthodes passives	23
1.1.4.1 Structure from motion	23
1.1.4.2 Reconstruction à partir d’images	24
1.1.5 Méthodes actives	26
1.1.5.1 Scanner 3D par balayage de plan laser	26
1.1.5.2 Caméra temps-de-vol	27
1.1.5.3 Caméra à lumière structurée	28
1.1.6 Étalonnage des capteurs à lumière structurée de type grand-public	30
1.1.6.1 Introduction	30
1.1.6.2 Paramètres intrinsèques	31
1.1.6.3 Méthodes d’étalonnage	33
2 Acquisition de la couleur	36
2.1 Introduction	36
2.2 L’apparence	36
2.2.1 La radiométrie	36
2.2.1.1 L’angle solide	36
2.2.2 La photométrie	36
2.2.2.1 Le flux énergétique	37
2.2.2.2 L’intensité énergétique	37
2.2.2.3 La luminance énergétique	37
2.2.2.4 L’éclairement énergétique	38
2.2.2.5 La fonction de distribution de réflectance bidirectionnelle	38
2.2.3 L’équation de rendu	38
2.2.4 Les propriétés de la BRDF	38

---

2.3	La géométrie de la réflexion . . . . .	39
2.4	Les modèles de réflexion . . . . .	41
2.4.1	La réflexion ambiante . . . . .	41
2.4.2	La réflexion spéculaire . . . . .	41
2.4.3	La réflexion diffuse . . . . .	43
2.5	Quelques modèles de BRDF . . . . .	43
2.5.1	Modèles empiriques . . . . .	43
2.5.1.1	Modèle de LAMBERT . . . . .	43
2.5.1.2	Modèle de PHONG . . . . .	44
2.5.2	Modèles analytiques . . . . .	44
2.5.2.1	Modèle de Ward . . . . .	44
2.5.2.2	Modèle de COOK-TORRANCE . . . . .	44
2.6	Dispositif de mesure de BRDF . . . . .	48
2.6.1	Gonioréfectomètre classique . . . . .	48
2.6.2	Gonioréfectomètre basé sur l'image . . . . .	48
<b>II</b>	<b>Étalonnage</b>	<b>50</b>
1	Introduction . . . . .	51
2	Étalonnage 2D . . . . .	52
3	Étalonnage de la profondeur . . . . .	52
3.1	Acquisition des données . . . . .	53
3.2	Définition de la grille et remplissage des nœuds . . . . .	54
3.2.1	Définition de la grille . . . . .	54
3.2.2	Remplissage des nœuds . . . . .	55
4	Application de la correction . . . . .	55
5	Setup matériel . . . . .	57
5.1	1er prototype . . . . .	57
5.2	Setup final . . . . .	57
6	Résultats et validation de la méthode . . . . .	60
6.1	Domaine d'étalonnage . . . . .	60
6.2	Étalonnage 2D . . . . .	60
6.3	Étalonnage de la profondeur . . . . .	62
6.3.1	L'erreur globale . . . . .	63
6.3.2	L'erreur locale . . . . .	63
6.3.3	Résultats . . . . .	63
6.3.4	Quelques notes sur la précision de la méthode . . . . .	66
6.4	Conclusion . . . . .	67
<b>III</b>	<b>Acquisition de la couleur</b>	<b>70</b>
1	Introduction . . . . .	71
2	Setup matériel . . . . .	71
3	Implémentation . . . . .	72
3.1	Algorithme . . . . .	73
3.1.1	Le recalage . . . . .	73
3.1.1.1	Principe . . . . .	73

---

3.1.1.2	L'algorithme ICP . . . . .	75
3.1.1.3	Algorithme . . . . .	80
3.1.2	Filtrage . . . . .	81
3.1.3	Estimation des normales . . . . .	81
3.1.4	Estimation des différents paramètres de la ou des BRDF choisie(s) . . . . .	83
3.2	Algorithme . . . . .	84
3.2.1	Résultats Cook-Torrance . . . . .	84
3.2.2	Résultats Ward . . . . .	85
3.2.3	Résultats Kelemen et Szirmay-Kalos . . . . .	87
	<b>Conclusion Générale . . . . .</b>	<b>90</b>
	<b>Bibliographie . . . . .</b>	<b>99</b>
	Publications et communications relatives à ce travail . . . . .	100

## Table des figures

1	Un sténopé. $f$ représente la distance focale. . . . .	19
2	Exemple de distorsion radiale. Les lignes en rouges montrent la courbure de lignes supposées être droites sur l'image. . . . .	22
3	Le point $M$ est "vu" par deux caméras $C_0$ et $C_1$ , avec ayant les centres $c_0$ et $c_1$ respectivement. $d_i$ est la distance entre $c_1$ et le point $M$ . $\hat{m}_i$ est la direction du rayon partant de $c_i$ et passant par $m_i$ . Les deux rayons s'intersecte au point $M$ . $(R, t)$ est la rototranslation pour passer de $C_0$ à $C_1$ . . . . .	22
4	Pipeline général des algorithmes reconstruction à partir d'images. [FH <sup>+</sup> 15]	25
5	Exemple d'un scanner par balayage de plan laser. . . . .	26
6	La caméra Kinect v2 est un exemple de caméra temps de vol grand public. La caméra est posée sur une feuille A4 afin d'avoir une idée sur sa taille. . . . .	27
7	L'impulsion de phase dans les caméra temps de vol. Un <i>mur de lumière</i> est émis, puis il prend la forme de l'objet qui le réfléchit vers le capteur. . . . .	28
8	Exemple d'un scanner 3D à lumière structurée. . . . .	29
9	Exemples de caméras 3D à lumière structurée grand public photographiées sur des feuilles A4 pour avoir une idée sur leur taille. En haut à gauche Microsoft Kinect pour XBox360, en haut à droite Occipital Structure Sensor et en bas Intel RealSense F200 . . . . .	30
10	Caméra Intel RealSense SR300 RGB-D. Les capteurs IR (gauche) et RVB (centre) sont clairement visibles sur l'image. Le projecteur IR est caché par la face avant teintée du boîtier. . . . .	31
11	Schéma représentatif de la notion d'angle solide. . . . .	37
12	Géométrie de la BRDF [Noe99] . . . . .	39
13	(a) Angles utiles pour une BRDF isotrope. (b) Angles utiles pour une BRDF anisotrope . . . . .	40
14	Géométrie de la réflexion : un observateur regarde un point $P$ sur une surface suivant la direction $V$ . $L$ est la direction de la lumière et $H$ la direction spéculaire . . . . .	40
15	Réflexion ambiante : la lumière provient uniformément de l'environnement puis elle est reflétée de façon équitable dans toutes les directions. . . . .	41
16	Réflexion spéculaire dans le cas d'un miroir lisse parfait : chaque rayon incident est réfléchi par la surface de telle sorte que l'angle entre le rayon incident et la normale à la surface est égal à l'angle entre le rayon réfléchi et la normale. . . . .	42
17	Réflexion spéculaire dans le cas d'un objet spéculaire réel : la lumière est réfléchié dans un cône centré sur la direction définie par les lois de Snell-Descartes. . . . .	42
18	Réflexion diffuse : la lumière est réfléchié dans plusieurs directions. . . . .	43
19	Distribution de Beckmann (a) $m = 0.2$ (b) $m = 0.6$ [CT82]. . . . .	46
20	Ombrage de la lumière incidente. . . . .	47
21	Masquage de la lumière réfléchié. . . . .	47
22	Gonioréfectomètre classique.[War92] . . . . .	48
23	Gonioréfectomètre basé sur l'image.[War92] . . . . .	49
24	Distorsion dans le SR300 IR. Le panneau avec le motif est rectangulaire. . . . .	51
25	Le plan d'étalonnage et sa configuration sur le CMM. . . . .	53

26	Coefficient de correction pour un point donné $P$ : la profondeur réelle du plan d'étalonnage $tD(P)$ divisée par la coordonnée $z$ de $P$ renvoyée par le capteur $sD(P)$ . . . . .	54
27	Vue de dessus de l'espace 3D nouvellement défini, $IJK$ (vue de dessus). . .	56
28	Notre 1er prototype. . . . .	57
29	Schéma de fonctionnement du prototype. . . . .	58
30	Dessus : le plan d'étalonnage se trouvant sur le "panneau intérieur" de la MMT. En bas : la sonde mécanique utilisée pour vérifier l'orthogonalité du plan avec l'axe $Y$ de la MMT. . . . .	59
31	Un alignement réel réussi ; nous avons utilisé des cercles verts super-posés sur le flux IR pour mettre en évidence les coins alignés. . . . .	59
32	Sur la gauche, une image du damier avant d'appliquer la correction. Sur la droite, la même image après correction de la distorsion. Les lignes rouges droites montrent l'effet de distorsion. . . . .	61
33	La sphère étalon utilisée dans notre processus de validation. Elle a un diamètre de 50.80 mm (2 pouces). . . . .	62
34	La sphère d'étalonnage capturée sur différentes positions à l'intérieur du domaine de l'étalonnage. La ligne bleue correspond à l'axe $Z$ du capteur. .	64
35	L'erreur globale et l'erreur locale du SDK par rapport à notre méthode. Nous avons moyenné 3 expériences. . . . .	65
36	Gauche : vue de face et de dessus d'un nuage de points (surface plane) avant correction. A droite, le même plan après correction en utilisant notre méthode. . . . .	66
37	Boite à lumière. . . . .	72
38	Environnement de la prise de vue : le modèle et le couple capteur/lumière ponctuelle. . . . .	73
39	Exemple de captures sous différents points de vue avec et sans lumières . .	74
40	Interface graphique du programme principal . . . . .	75
41	Enchaînement de notre algorithme . . . . .	76
42	Exemple de Recalage 3D entre deux captures. Nous considérons le repère $R1$ de la 1ère prise de vue comme référence et il faut trouver la rotation $R$ et la translation $t$ (matrice de transformation) pour passer du repère $R2$ vers le repère $R1$ . . . . .	77
43	Exemple d'un pass-through filter avec la définition d'intervalles bornés pour les trois axes $X$ , $Y$ et $Z$ . Ainsi, on définit une boîte dans l'espace de telle sorte que chaque point se trouvant à l'extérieur de la boîte est éliminé. . .	78
44	Elimination par critère de voisinage sphérique. Dans cet exemple, on élimine les points qui ont moins de deux voisins sur un rayon $R$ . Le point à gauche (rouge) est éliminé alors que celui de droite (vert) reste. . . . .	78
45	Sous-échantillonnage par grille de voxels. Chaque sous-nuage de points appartenant à un voxel est remplacé par son barycentre. . . . .	79
46	Recherche de correspondants : A gauche, recherche du point le plus proche et à droite la projection selon le point de vue de l'autre nuage. . . . .	80
47	Métriques d'erreur. A gauche la métrique point-à-point et à droite la métrique point-à-plan [HIT <sup>+</sup> 15]. . . . .	81
48	Exemple de nuages de points avant recalage en haut (4 nuages), et exemple après recalage en bas (28 nuages). . . . .	82

49	Exemple de filtrage avec un minimum de 15 voisins sur un rayon de 1 mm sur le nuage final (recalé). A droite le nuage avant et à gauche le nuage après filtrage . . . . .	82
50	Exemple de nuage de points avec les normales estimées (affichés dans le logiciel [urlr]) . . . . .	83
51	(a) Rendu direct avec la couleur des captures sans lumières, (b) Rendu avec les paramètres estimés de Cook-Torrance (c) Image Réelle . . . . .	85
52	(a) Rendu direct avec la couleur des captures sans lumières, (b) Rendu avec les paramètres estimés de Ward (c) Image Réelle . . . . .	85
53	(a) Rendu direct avec la couleur des captures sans lumières, (b) Rendu avec les paramètres estimés de Kelemen et Szirmay-Kalos (c) Image Réelle . . .	87

# Liste des tableaux

1	Les valeurs de l'étalonnage par damier comparées aux valeurs du SDK. . .	60
2	Evaluation de l'erreur globale. . . . .	64
3	Modèle de Cook-Torrance : résultats obtenus pour différentes valeurs initiales $m_0$ et $k_0$ . $m$ et $k$ sont les moyennes de chaque paramètre sur les points optimisés. . . . .	84
4	Modèle de Ward : résultats obtenus pour différentes valeurs initiales $m_x0$ , $m_y0$ et $k_0$ . $m_x$ , $m_y$ et $k$ sont les moyennes de chaque paramètre sur les points optimisés. . . . .	86
5	Modèle de Kelemen et Szirmay-Kalos : résultats obtenus pour différentes valeurs initiales $m0$ et $k0$ . $m$ et $k$ sont les moyennes de chaque paramètre sur les points optimisés. . . . .	88

# Introduction générale

# 1 Contexte

L'imagerie numérique, de la synthèse d'images à la vision par ordinateur connaît une forte évolution, due entre autres facteurs à la démocratisation et au succès commercial des caméras 3D. Dans le même contexte, l'impression 3D grand public, qui est en train de vivre un essor fulgurant, contribue à la forte demande sur ce type de caméra pour les besoins de la numérisation 3D.

Cette évolution a commencé avec l'arrivée de la Kinect [urlo] en Novembre 2010. Elle a connu non seulement un réel succès commercial, mais a aussi suscité un grand intérêt dans la communauté scientifique : plus de 3 000 articles scientifiques ont été publiés en rapport avec la Kinect [BMNK13], dont une bonne partie traite de la numérisation 3D. En 2014, la technologie Kinect fait une nouvelle fois l'objet d'un cours à Siggraph [urlp]). De plus, de nouveaux périphériques sont étudiés et proposés régulièrement dans ce même contexte, comme le Structure Sensor [urlv], apportant la capture 3D aux tablettes, ou encore le capteur DepthSense au coeur des caméras SoftKinetic, qui fut racheté par Sony [urlu].

Ainsi, les caméras 3D même les plus modestes embarquent un nombre grandissant de capteurs permettant le développement de nouvelles applications et de méthodes innovantes liées à la synthèse d'images : calcul de la profondeur des objets filmés, reconstruction géométrique, extraction de textures et de caractéristiques optiques ... Ces nouveaux « dispositifs d'acquisition » sont aujourd'hui courants, ils existent sous la forme d'appareils portables et peuvent même être intégrés à des smartphones [urlw], rendant à terme la base de matériels installée potentiellement très importante.

Si la communauté scientifique s'intéresse depuis longtemps à la problématique de la reconstruction 3D, il n'en va pas de même pour ce qui concerne l'acquisition des caractéristiques photométriques représentées par les BRDF (Bidirectional Reflectance Distribution Function) et les textures. Cette thèse se place précisément dans ce contexte.

Le champ des applications basées sur la capture 3D est très large, incluant entre autres, la numérisation 3D, la réalité augmentée, la cartographie / localisation en simultané (*simultaneous localization and mapping*), le suivi d'objets ou encore la reconnaissance de gestes. Cette thèse sera focalisée dans les contextes de la numérisation des caractéristiques photométriques. La numérisation géométrique 3D, réservée jusque-là aux industriels à cause de la taille et le prix des scanners 3D, devient plus accessible. La réalité augmentée consiste à intégrer des images ou données virtuelles dans des vidéos capturées en temps réel. Dans le domaine de la mode par exemple, cette technologie a permis la création de cabines d'essayage virtuel, comme par exemple VirtualFashion [urlx]. Ceci montre qu'en plus de l'acquisition de la géométrie, la simulation visuelle du comportement des matériaux (propriétés de réflexion, interactions, textures) a une influence majeure.

## 2 Objectif

L'objectif de cette thèse est d'acquérir et de maîtriser un savoir-faire dans le domaine de la capture/acquisition de modèles 3D en particulier sur l'aspect rendu réaliste. La réalisation d'un scanner 3D à partir d'une caméra RGB-D fait partie de l'objectif.

Lors de la phase d'acquisition, en particulier pour un dispositif portable, on est confronté à deux problèmes principaux, le problème lié au référentiel de chaque capture et le rendu final de l'objet reconstruit.

### 2.1 Acquisition

Une des techniques qui traite du problème de référentiel est la Cartographie et Localisation Simultanée ou SLAM (Simultaneous Localization And Mapping). L'algorithme KinectFusion [INK<sup>+</sup>] et ses dérivés constituent une première réponse.

De plus, la fusion globale des données disponibles (gyroscopes, caméra infrarouge, microphone ..) peut entrer en jeu dans ce processus d'étalonnage/stabilisation. Par exemple, l'adjonction d'un gyroscope aux capteurs RGB-D à obturateur déroulant rolling shutter permet d'améliorer la précision des cartes de disparités depth maps [OFT13]. Dans ce même contexte, Lee et al. ont travaillé sur la correction de la géométrie des modèles acquis [LJB<sup>+</sup>13], en analysant l'ombrage dans des images infrarouges des modèles.

### 2.2 Rendu

L'acquisition des propriétés de réflexion, modélisées par des fonctions bidirectionnelles de distribution de la réflectance (BRDF), ainsi que celle des textures caractéristiques des objets, sont très importantes comme nous l'avons mentionné dans le contexte de ce sujet. Cependant, il s'agit d'un domaine de recherche récent, où peu de travaux de recherches ont été menés jusqu'à présent. Knecht et al. ont proposé une méthode d'estimation de BRDFs à l'aide d'une Kinect [KTTW12]. Dans une autre voie, Calian et al. ont présenté une sonde de mesure dédiée aux périphériques mobiles [CMNK13].

## 3 Contributions

Nous avons proposé une nouvelle méthode de calibration de capteurs à lumière structurée qui améliore les résultats par rapport aux paramètres du constructeur. Les essais ont été menés sur une camera Intel RealSense SR300, mais la méthode n'est pas restreinte à cette capteur car nous faisons intervenir des paramètres communs à tout type de caméra utilisant la même technologie. Nous avons présenté cette méthode dans la conférence internationale WSCG 2018.

## 4 Organisation de la thèse

Dans cette thèse, nous nous intéressons à deux problématiques principales : la numérisation 3D via des capteurs de type grand public ainsi que le rendu des objets numérisés.

Concernant la numérisation 3D, l'acquisition de la géométrie par un dispositif grand public présente des inconvénients liés à la qualité des scans ainsi que la précision si nous comparons avec les solutions de grade industriel.

Quant au rendu, il s'agit de proposer un modèle adéquat à l'objet numérisé en profitant des diverses informations obtenues avec le capteur 3D. Ces informations devraient permettre de réduire le nombre des paramètres à estimer dans le modèle de rendu.

Ce manuscrit est composé de cinq grandes parties :

- Une introduction générale qui présente le contexte, la problématique ainsi que l'organisation du manuscrit.
- Un état d'art divisé en deux parties. La première comporte une présentation générale des différentes techniques de numérisation 3D en se focalisant sur les capteurs à lumière structurée ainsi qu'une revue du pipeline typique de scan 3D. La deuxième contient une introduction aux BRDF avec les notions de bases et quelques variantes que nous avons pu explorer dans nos expérimentations.
- Une nouvelle méthode d'étalonnage de capteurs à lumière structurée qui repose sur l'utilisation d'une machine à mesure tridimensionnelle (MMT) dont la grande précision nous a permis d'améliorer nos résultats par rapport à l'étalonnage usiné.
- Une méthode basée sur l'utilisation d'un capteur 3D avec une source de lumière attachée de manière rigide, afin de simplifier l'estimation de BRDF.
- Une conclusion générale.

Première partie

État de l'art

# 1 Acquisition de la géométrie

## 1.1 Techniques d'acquisition de type optique sans-contact

Pour numériser un objet ou une scène, plusieurs techniques d'acquisition 3D existent. Le but de ce chapitre est de fournir une classification par contexte d'utilisation des méthodes d'acquisition optique sans-contact. Par exemple, quel type de méthode choisir pour la numérisation d'une surface non texturée ?

Généralement, les méthodes d'acquisition de type optique sans-contact peuvent être séparées en deux branches, voir [LT09], [MVG<sup>+</sup>10] et [Cur00] :

- Les *méthodes actives* utilisent un éclairage contrôlé pour illuminer l'objet à numériser.
- Les *méthodes passives* reposent uniquement sur l'éclairage ambiant lors du processus d'acquisition.

Plusieurs auteurs apportent des études sur le sujet. Dans son papier, Blais [Bla04] donne une vue globale sur les différentes approches qui existent en se focalisant plus particulièrement sur les solutions commerciales valables au moment de la publication du papier. Pour chaque système, il fournit les informations sur la technologie, la portée et la précision lorsqu'elles sont valables. Curless a un cours SIGGRAPH sur les méthodes d'acquisition actives [Cur00].

Zhang et al. ont passé en revue les différentes techniques de type shape from shading [ZTCS99], en évaluant six variantes de cette classe d'algorithmes.

Pour les techniques de lumière structurée, nous pouvons citer le papier de Salvi et al. [SFPL10], qui présente les méthodes de codification et les différents patrons utilisés dans ce type de numérisation 3D.

D'autres papiers classent les méthodes de numérisation 3D plutôt par domaine d'application, Musialski et al. présentent les techniques de reconstruction urbaine [MWA<sup>+</sup>13]. Quant à González-Aguilera et al., ils étudient les techniques de numérisation 3D et de modélisation de caves [GAMNGL<sup>+</sup>09].

### 1.1.1 Le modèle sténopé

Un sténopé consiste en une boîte fermée, avec du papier photosensible sur l'une de ses parois intérieures. Du côté opposé, un trou minuscule permet aux rayons de lumière de pénétrer à l'intérieur de la boîte afin de former une image sur le papier photosensible [MVG<sup>+</sup>10]. Nous pouvons imaginer un trou produit à l'aide de la pointe d'une aiguille, d'où le nom anglais *pinhole camera*.

En pratique, afin de laisser passer une plus grande quantité de lumière, nous pourrions envisager un trou avec un diamètre plus important, mais ceci produira des images floues.

Cet effet est dû au fait que les points 3D réels produisent plusieurs projections sur le papier photosensible.

L'utilité du modèle sténopé vient du fait qu'il permet de décrire la géométrie de formation d'image pour une caméra, malgré sa simplicité. En effet, c'est le modèle le plus basique pour la représentation d'une caméra [HZ05].

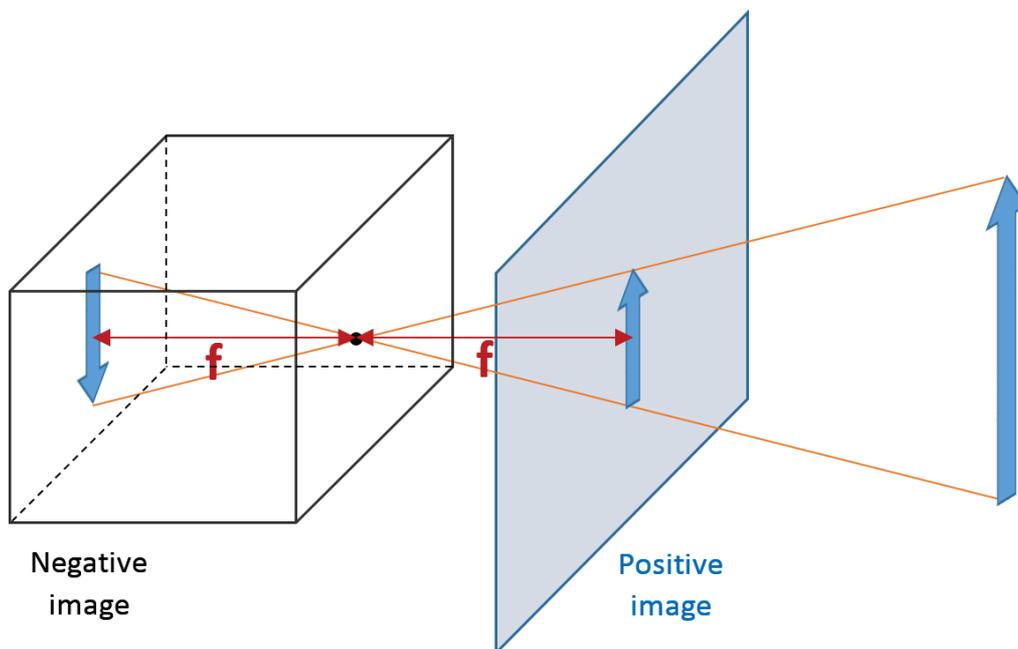


FIGURE 1 – Un sténopé.  $f$  représente la distance focale.

La figure 1 présente la composition d'un sténopé. Le trou laissant passer la lumière est appelé *le centre optique* et l'image positive est appelée *le plan image* ou *le plan focal*. La ligne perpendiculaire au plan image et passant par le centre optique est appelée *l'axe principal*. L'intersection entre l'axe principal et le plan image est appelée *le point principal*.

Considérons un repère Euclidien placé sur le centre optique du sténopé, de telle sorte que l'axe  $Z$  du repère Euclidien est aussi l'axe principal de la caméra.

En utilisant le principe des triangles similaires, un point de l'espace  $M = (X, Y, Z)^t$  a pour image le point  $m = (fX/Z, fY/Z, f)^t$  dans le plan image. L'espace image étant plan, la troisième coordonnée peut être ignorée. Du coup la transformation suivante, définie de  $\mathbb{R}^3$  dans  $\mathbb{R}^2$ , décrit la projection centrale du monde vers les coordonnées images :

$$(X \ Y \ Z)^t \rightarrow \left( \frac{fX}{Z} \ \frac{fY}{Z} \right)^t \quad (1)$$

En général, le repère sur le plan image n'est pas centré sur le point principal. Ainsi, si  $p = (p_x, p_y)$  est le point principal, la transformation finale s'écrit :

$$(X \ Y \ Z)^t \rightarrow \left( \frac{fX}{Z} + p_x \ \frac{fY}{Z} + p_y \right)^t \quad (2)$$

### 1.1.2 Les paramètres intrinsèques et extrinsèques

Considérons la matrice  $K$  définie par :

$$K = \begin{pmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

Alors, la transformation 2 peut être réécrite de la manière suivante :

$$\begin{pmatrix} fX + Zp_x \\ fY + Zp_y \\ Z \end{pmatrix} = \begin{pmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = K[I|0]M \quad (4)$$

La matrice  $K$  dépend de deux paramètres uniquement, la distance focale et le point principal. Ces deux paramètres sont intrinsèques à la caméra, d'où le nom donné à la matrice  $K$  : *matrice d'étalonnage intrinsèque* à la caméra [HZ05].

Dans le système d'équations 4, nous avons considéré le repère caméra comme repère de référence. Cette hypothèse est justifiable dans le cas d'un système composé d'une seule caméra, puisqu'elle simplifie le système d'équations en travaillant directement avec le repère de la caméra. Dans un système avec plusieurs caméras, nous choisissons le repère monde (de référence) indépendamment des repères des caméras.

Revenons au système 4. En choisissant le repère monde comme référence, nous considérons la matrice de transformation (rotation)  $R$  décrivant l'orientation du repère de la caméra et le point  $C$  la position de ce dernier. Nous dénotons  $M_{cam}$  comme la représentation dans le repère caméra d'un point  $M$ .  $M$  exprime les coordonnées du point dans le repère monde. Alors :

$$M_{cam} = R(M - C) \quad (5)$$

Ainsi, le système 4 devient :

$$m = KR[I| - C]M = K[R|t]M \quad (6)$$

Nous définissons la matrice  $P$  comme suit :

$$P = K[R|t] \quad (7)$$

La matrice  $R$  et le vecteur  $t$  sont appelées *les paramètres extrinsèques* de la caméra, avec  $t = -RC$ . La matrice  $P$  est appelée *la matrice d'étalonnage*.

L'opération de recherche des paramètres intrinsèques et extrinsèques d'une caméra s'appelle *l'étalonnage de caméra*. L'étalonnage peut être uniquement intrinsèque, en se contentant uniquement des paramètres intrinsèques de la caméra, il peut être extrinsèque ou les deux en même temps. Nous allons en parler plus en détail dans la section « Étalonnage ».

En pratique, les lentilles utilisées présentent des distorsions de type géométrique. Ces distorsions sont dues essentiellement à des défauts d'asymétrie ou d'agencement des lentilles dans la caméra. Ainsi, les positions des points sur le plan image décrites par le système 4 ne sont pas correctes et le système doit prendre en compte la distorsion géométrique [WCH<sup>+</sup>92].

Soit  $X_c = (x_c, y_c, z_c)^T$  un point et  $P_c = (u_c, v_c)^T$  sa projection sur le plan image. Notons  $X_n = (x_c/z_c, y_c/z_c)$  ce que les auteurs du papier [HKH<sup>+</sup>12] appellent point normalisé.

Les distorsions géométriques (radiale et tangentielle) s'appliquent comme suit :

$$X_g = \begin{pmatrix} 2k_3x_ny_n + k_4(r^2 + 2x_n^2) \\ k_3(r^2 + 2y_n^2) + 2k_4x_ny_n \end{pmatrix} \quad (8)$$

$$X_k = (1 + k_1r^2 + k_2r^4 + k_5r^6)X_n + X_g \quad (9)$$

Avec  $r^2 = x_n^2 + y_n^2$  et  $k_c = [k_1, ..k_5]$  le vecteur des coefficients de distorsion.

Les coordonnées sur le plan image s'écrivent [HKH<sup>+</sup>12] :

$$u_c = f_{cx}x_k + u_{0c} \quad (10)$$

$$v_c = f_{cy}y_k + v_{0c} \quad (11)$$

### 1.1.3 La triangulation stéréo passive

À partir d'au moins deux images de la même scène prises sous différents points de vue, et les paramètres des caméras ayant capturées des images, nous pourrions récupérer la coordonnée 3D d'un point de la scène en utilisant ses projections dans les images.

Considérons deux caméras  $C_0$  et  $C_1$  avec une pose relative inconnue. Soit  $M$  un point physique 3D avec  $m_0$  et  $m_1$  ses deux projections 2D en  $C_0$  et  $C_1$  respectivement. Comme la rototranslation  $(R, t)$  est inconnue entre les deux caméras, nous pouvons prendre  $C_0$  comme référence du système sans perte de généralité. Ainsi, nous considérons que la caméra  $C_0$  représente le repère référence (monde). Voir la figure 3.

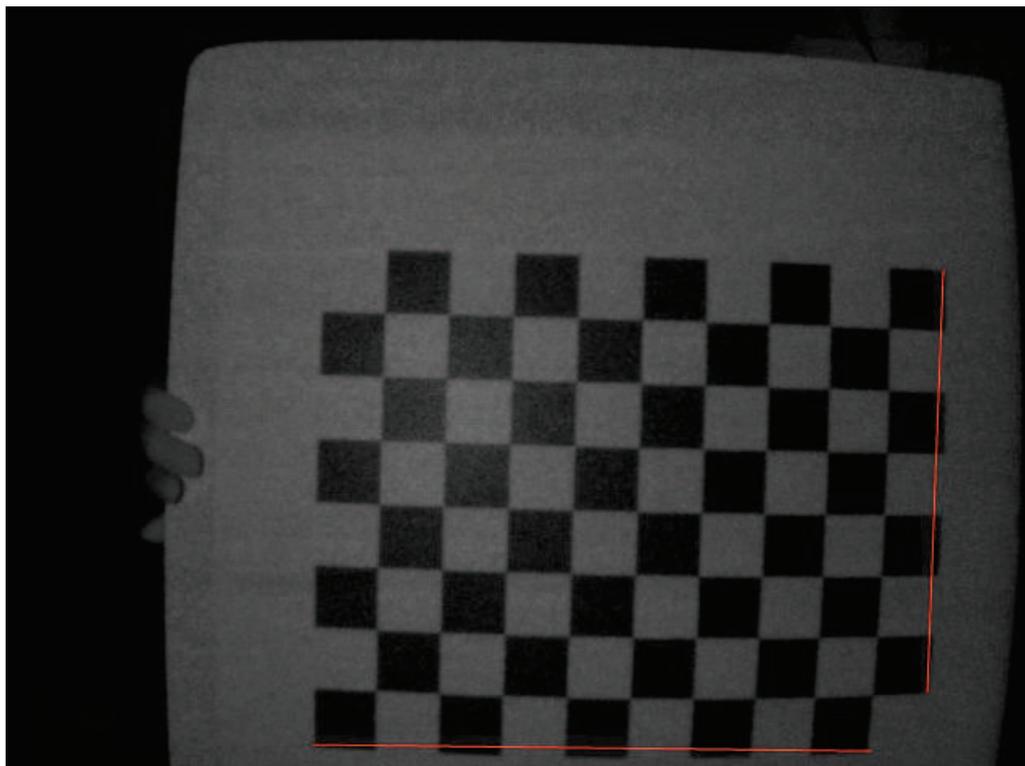


FIGURE 2 – Exemple de distorsion radiale. Les lignes en rouges montrent la courbure de lignes supposées être droites sur l'image.

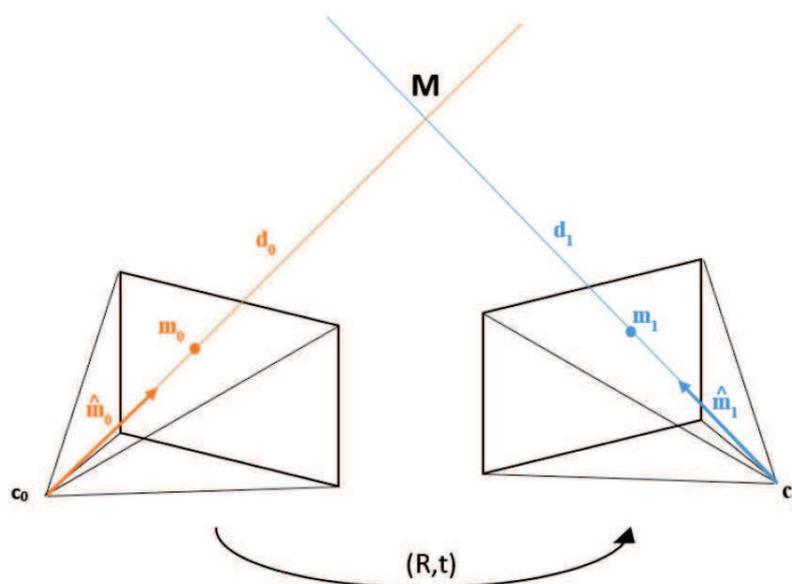


FIGURE 3 – Le point  $M$  est "vu" par deux caméras  $C_0$  et  $C_1$ , avec ayant les centres  $c_0$  et  $c_1$  respectivement.  $d_i$  est la distance entre  $c_i$  et le point  $M$ .  $\hat{m}_i$  est la direction du rayon partant de  $c_i$  et passant par  $m_i$ . Les deux rayons s'intersecte au point  $M$ .  $(R, t)$  est la rototranslation pour passer de  $C_0$  à  $C_1$ .

Sous les précédentes conditions, et en notant  $\hat{m}_j = K^{-1}m_j$  la direction de chaque rayon :

$$d_1\hat{m}_1 = m_1 = Rm_0 + t = R(d_0\hat{m}_0) + t \quad (12)$$

$$d_1[t]_x\hat{m}_1 = d_0[t]_xR\hat{m}_0 \quad (13)$$

$$d_1\hat{m}_1^T[t]_x\hat{m}_1 = d_0\hat{m}_1^T[t]_xR\hat{m}_0 = 0 \quad (14)$$

$$\hat{m}_1^T([t]_xR)\hat{m}_0 = 0 \quad (15)$$

La matrice  $E = [t]_xR$  est appelée la *matrice essentielle* [Sze10]. Cette matrice permet d'établir la relation entre deux vues.

### 1.1.4 Méthodes passives

**1.1.4.1 Structure from motion** Les algorithmes Structure from motion (SfM) sont une famille d'algorithmes de reconstruction 3D qui génèrent un modèle 3D à partir d'un ensemble d'images non structurées, c'est-à-dire des images sans connaissances préalables des positions et des paramètres de caméra lors des captures. En plus du modèle 3D, ces méthodes permettent de calculer les positions des prises de vues ainsi que les paramètres intrinsèques de ou des caméras utilisées pour les captures des images d'entrée. Les objets numérisables par ce type de techniques varient de petits objets à des villes entières. En effet, il existe des algorithmes SfM pouvant prendre plusieurs milliers d'images en entrée [AFS<sup>+</sup>11].

Les étapes principales d'un algorithme de type SfM sont :

- la détection 2D des points caractéristiques dans toutes les images ;
- la recherche de correspondances (dans l'ensemble des points caractéristiques) entre les images ;
- le filtrage des correspondants ;
- la reconstruction 3D.

Parmi les méthodes les plus utilisées dans la détection 2D de points caractéristiques et la recherche de correspondants dans les algorithmes SfM, nous pouvons citer SIFT (Scalable-Invariant Feature Transform) et SURF (Speeded Up Robust Features) voir [HP10]. SIFT est un algorithme de détection locale des zones d'intérêts dans les images 2D, caractérisé par sa robustesse contre les changements d'échelle et des rotations [Low04]. Quant à la méthode SURF, elle utilise les ondelettes de Haar et les images intégrales pour des résultats plus robuste que l'algorithme SIFT selon ses auteurs [BTVG06].

Pour le filtrage, l'algorithme RANSAC (RANdom SAmple Consensus) [FB87] est utilisé pour éliminer les faux correspondants.

La reconstruction peut être réalisée via la géométrie épipolaire que nous avons présentée un peu plus haut.

Un algorithme de type SfM retournent les paramètres de la caméra en sortie, ainsi, il peut être considéré comme une méthode d'étalonnage.

Les méthodes de type SfM ont le grand avantage de ne prendre qu'un ensemble d'images en entrée sans avoir besoin d'autres informations sur les paramètres de caméra ou les positions de prises de vue. Par contre, la dépendance totale des algorithmes de détections de points caractéristiques exigent d'avoir des scènes ou des objets avec des zones d'intérêts "détectables". Du coup, les objets lisses ou faiblement texturés peuvent présenter des difficultés de reconstruction.

**1.1.4.2 Reconstruction à partir d'images** Le but de la reconstruction à partir d'images est de générer un modèle 3D, d'un objet ou d'une scène, à partir d'un ensemble d'images calibrées et capturées sous des conditions bien connues. La qualité de la reconstruction dépend directement de la qualité des images d'entrées, les positions de prises de vues correspondantes et les paramètres intrinsèques des caméras utilisées.

Seitz et al. [SCD<sup>+</sup>06] ont utilisé des critères de classification pour les algorithmes de type reconstruction à partir d'images. Voici un bref résumé :

- **L'initialisation** : un ensemble d'images calibrées n'est pas suffisant pour la reconstruction d'un modèle 3D. Ainsi, d'autres contraintes doivent être prises en compte, comme une délimitation "grossière" de la scène via une boîte englobante ou un volume par exemple.
- **L'évaluation de la photo-consistance** : mesure le niveau de similitude entre les pixels correspondants dans l'ensemble des images. L'évaluation peut être effectuée dans l'espace image ou l'espace scène. D'un côté, les mesures de l'espace scène consistent à projeter un élément de géométrie (que ce soit un point, un volume ou un patch), vers les images d'entrée afin d'extraire les similarités entre les différentes projections. D'un autre côté, les mesures dans l'espace image utilisent l'estimation de la géométrie de la scène pour passer d'un point de vue à l'autre via des transformations de type déformation.
- **Le modèle de visibilité** : aide à résoudre les problèmes de visibilité/occlusions entre les différents points de vue. En effet, ce modèle extrait un sous-ensemble d'images partageant des points communs visibles afin de pouvoir évaluer la photo-consistance. Ceci peut être appliqué en utilisant l'estimation courante pour deviner que tel point est visible dans telle vue, ou bien en appliquant l'estimation de la photo-consistance uniquement sur un sous-ensemble de caméras proches. D'autres

techniques existent et nous dirigeons le lecteur vers le papier [SCD<sup>+</sup>06] pour plus de détails.

- **Shape prior** : permet d'ajouter davantage de contraintes au modèle, que ce soit localement ou globalement, afin d'arriver à la reconstruction désirée. En effet, les contraintes provenant de multiples images d'entrée sont moins efficaces quand il s'agit de parties faiblement texturées de la scène.
- **La reconstruction** : il y a quatre sorties possibles pour un algorithme de type reconstruction à partir d'images [FH<sup>+</sup>15], volume, carte de disparité, nuage de points ou maillage raffiné. Nous notons que pour le type de sortie nuage de points, une méthode de rendu colorant les points du nuage permet de donner un effet de "texture" au modèle final sans avoir à générer un maillage.

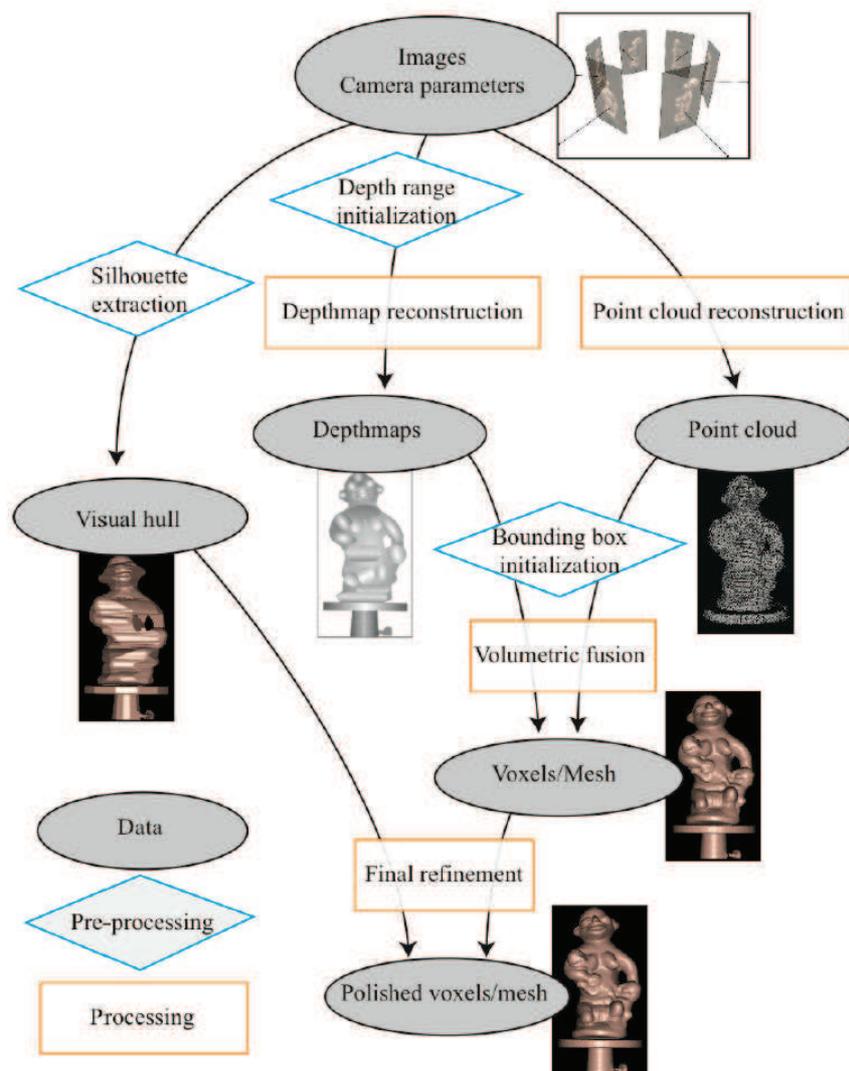


FIGURE 4 – Pipeline général des algorithmes reconstruction à partir d'images. [FH<sup>+</sup>15]

Un pipeline général [FH<sup>+</sup>15] pour les algorithmes reconstruction à partir d'images est

présenté dans la figure 4. L'étape de reconstruction peut comprendre plusieurs niveaux, par exemple, nous pouvons nous contenter d'un nuage de points ou bien continuer jusqu'à obtenir un volume ou un maillage raffiné.

### 1.1.5 Méthodes actives

**1.1.5.1 Scanner 3D par balayage de plan laser** Ce dispositif était devenu populaire dans les années 80 avec l'introduction de matrice CCD de qualité et à faible coût. La caméra est couplée à un projecteur laser qui projette une ligne laser sur l'objet à numériser. Un système mécanique déplace la ligne laser sur la surface de l'objet afin d'acquérir sa géométrie [LT09].

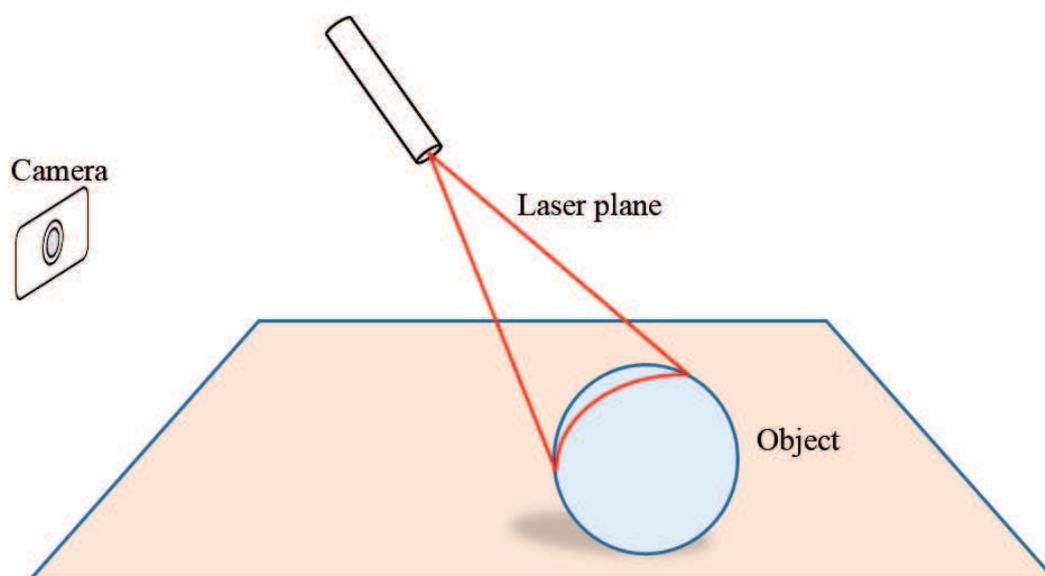


FIGURE 5 – Exemple d'un scanner par balayage de plan laser.

Ce type de système peut être utilisé pour numériser des objets de différentes tailles, cela va de petits objets à une capture complète d'un corps humain [url].

La résolution peut être aux alentours de un millimètre dans de petits périphériques grand public pour atteindre quelques dizaines de microns dans des solutions industrielles [url].

Malgré leur résolution très élevée, les scanners par balayage de plan laser souffrent de quelques faiblesses. En effet, ils sont sensibles à l'éclairage ambiant, ce qui les rend moins adaptés à un usage extérieur [Bla04]. Un autre point faible réside dans le fait que ce type de dispositif est incapable de capturer des scènes dynamiques. Ceci est dû au mouvement de la ligne laser qui doit balayer l'intégralité de la surface de l'objet et ce dernier doit

être immobile lors de la numérisation. La séparation entre la caméra et le laser engendre des difficultés de captures dues aux occlusions sur la surface de l'objet.

**1.1.5.2 Caméra temps-de-vol** Dans une caméra temps de vol, une impulsion lumineuse est lancée à partir d'un module d'illumination et un capteur image la reçoit en retour. La vitesse de la lumière est utilisée afin de déduire la distance de l'objet en mesurant le temps écoulé entre l'émission et la réception de l'impulsion. Ainsi, la profondeur  $P$  est obtenue par la formule suivante :

$$P = \frac{1}{2}c.T \quad (16)$$

Où  $c = 300000000 \text{ m/s}$  est la vitesse de la lumière et  $T$  la durée de l'aller-retour de l'impulsion.



FIGURE 6 – La caméra Kinect v2 est un exemple de caméra temps de vol grand public. La caméra est posée sur une feuille A4 afin d'avoir une idée sur sa taille.

Dans la plupart des caméras temps de vol, une image d'amplitude est capturée en même temps que l'image de profondeur. L'image d'amplitude mesure la quantité de lumière retournée et permet ainsi d'associer un facteur de fiabilité aux profondeurs mesurés.

Parmi les approches utilisées dans les systèmes temps de vol, nous pouvons citer la modulation continue et l'impulsion de phase [LNL<sup>+</sup>13].

- **La modulation continue** : c'est une fonction de corrélation, entre le signal émis décalé par un offset de phase et le signal incident ;
- **L'impulsion de phase** : il s'agit d'émettre une impulsion d'une quantité de lumière connue, et de couper le signal réfléchi par un obturateur avant qu'il

n'atteigne le capteur CCD. La "tranche" capturée par le module CCD reflète la forme de l'objet. L'impulsion émise par la caméra est appelée *mur de lumière* [KBKL09]. La figure 7 illustre le concept ;

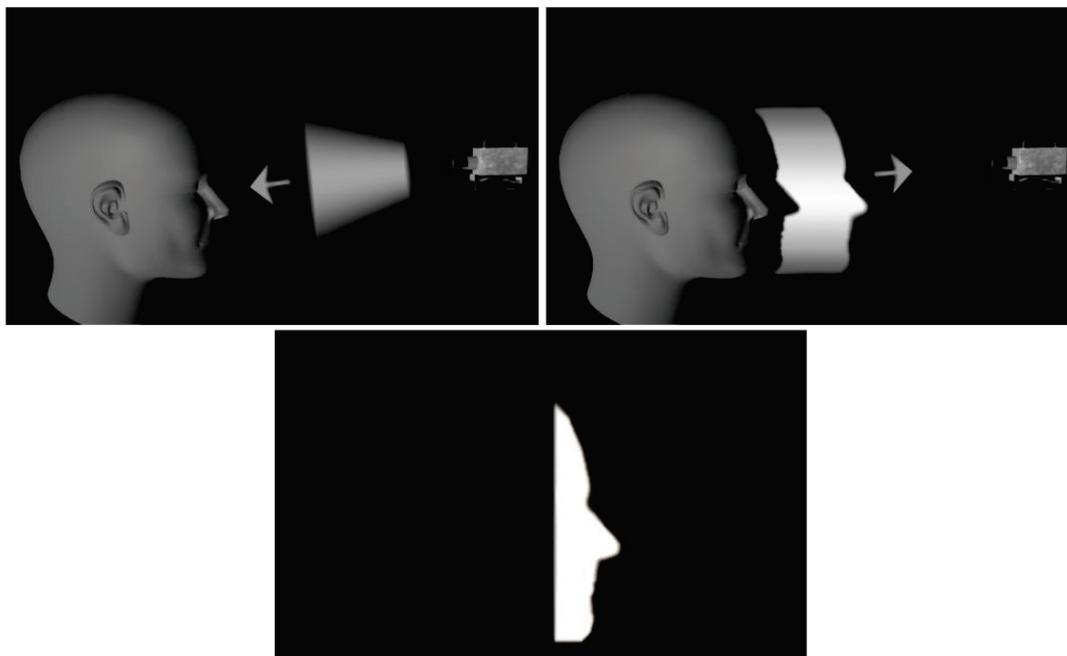


FIGURE 7 – L'impulsion de phase dans les caméra temps de vol. Un *mur de lumière* est émis, puis il prend la forme de l'objet qui le réfléchit vers le capteur. [KBKL09] [IY01].

Les caméras temps de vol repose sur des modules électroniques très sensibles. Ainsi, ces modules requièrent une bonne stabilité thermique. En pratique, plusieurs impulsions sont moyennées afin de réduire le bruit. La résolution de ce type de caméra peut aller de 1 cm jusqu'à 0,5 mm [Bla04].

Concernant leur portée, ces caméras sont capables de mesurer de grandes structures. En effet, ils peuvent atteindre plusieurs centaines de mètres [Bla04]. Ces caméras sont très rapides et permettent, contrairement aux scanners lasers, de mesurer la scène en une seule capture (impulsion). Par contre, leur résolution restent en deçà des scanners lasers.

**1.1.5.3 Caméra à lumière structurée** En contraste avec les scanners à balayage laser, le mouvement mécanique est remplacé par un patron (pattern) dans les caméra à lumière structurée. Un scanner 3D à lumière structurée est composé d'un projecteur et d'une ou plusieurs caméras pour capturer le patron (pattern) projeté sur l'objet ou la scène à scanner.

La correspondance entre les patrons codés et leur projection permet de déduire l'information 3D de la surface [SFPL10].

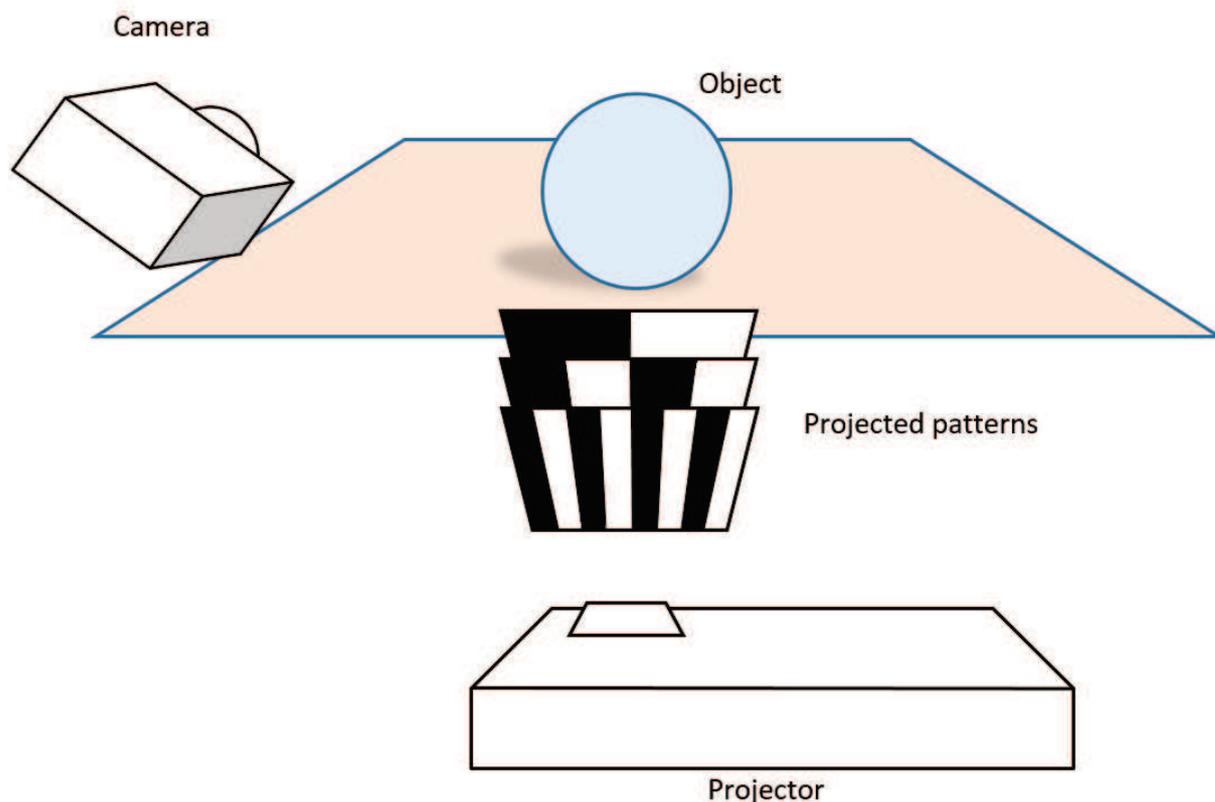


FIGURE 8 – Exemple d'un scanner 3D à lumière structurée.

Ce type de système peut être calibré avec un damier en utilisant la méthode de Zhang [Zha00]. Le processus est décrit en détail dans le cours de Lanman [LT09]. Des toolboxes sont aussi disponibles comme par exemple [urlc]. Ainsi en principe, la calibration d'un scanner 3D à lumière structurée reste très accessible car elle ne demande qu'un damier imprimée ou peint sur une surface plane ainsi qu'un logiciel dédié comme la toolbox qu'on vient de citer.

L'introduction du capteur 3D grand public Microsoft Kinect en 2010 a constitué la naissance d'une nouvelle catégorie de caméras 3D à lumière structurée. C'est un système très compact, de telle sorte qu'un projecteur infrarouge, une caméra infrarouge ainsi qu'une caméra RVB sont contenus dans un petit boîtier 9. Depuis, ce type de caméra est devenu très accessible et largement utilisé.

Les capteurs 3D à lumière structurée ont une portée qui varie de quelques centimètres à plusieurs mètres. La caméra Intel RealSense F200 opère sur un intervalle qui va de 20cm à 120cm [urla]. Quant à la caméra Microsoft Kinect v1, sa portée va de 50cm à 5m [KE12]. Les angles de vision sont généralement fournies par les constructeurs. Ainsi, la portée et les angles de vision permettent de calculer les dimensions de la "boîte" de vision (une pyramide tronquée) du capteur. Au final, une caméra à lumière structurée peut scanner de petits objets et aller jusqu'à de grands sujets comme une chambre par exemple.



FIGURE 9 – Exemples de caméras 3D à lumière structurée grand public photographiées sur des feuilles A4 pour avoir une idée sur leur taille. En haut à gauche Microsoft Kinect pour XBox360, en haut à droite Occipital Structure Sensor et en bas Intel RealSense F200

Comme indiqué dans le papier [KE12], l'erreur sur la profondeur accroit et la résolution de la profondeur diminue en s'éloignant de la caméra Kinect.

### 1.1.6 Étalonnage des capteurs à lumière structurée de type grand-public

**1.1.6.1 Introduction** Outre le fait de leur accessibilité à la fois en termes de coût et de facilité d'utilisation, les caméras à lumière structurées grand-public souffrent en général d'un problème majeur qui est la précision si nous les comparons aux solutions de grade industriel. En effet, les constructeurs utilisent généralement des méthodes d'étalonnage propriétaires avec leurs appareils ce qui conduit à des technologies semi-fermées, en particulier pour les utilisateurs expérimentés. Heureusement, certains fabricants fournissent un accès aux paramètres d'étalonnage à travers leurs SDK, comme Intel avec leurs modèles RealSense par exemple. Ainsi, un utilisateur averti peut expérimenter avec différentes valeurs de ces paramètres et les appliquer aux mesures brutes provenant du capteur, au lieu de travailler avec les fonctions de capture haut-niveau prédéfinies du SDK. De plus, l'utilisation d'une meilleure méthode d'étalonnage par rapport à celle du fabricant permet de remplacer les paramètres par des substituts plus précis. En fait, il y a eu beaucoup de tentatives réussies et prometteuses avec le capteur Kinect (voir la section sur les méthodes d'étalonnage pour quelques exemples).

En général, l'étalonnage d'une caméra consiste en deux étapes :

- L'étalonnage intrinsèque qui est le calcul des paramètres internes de la caméra tels que la distance focale, le centre optique et les distorsions radiales et tangentielles
- L'étalonnage extrinsèque qui correspond à la pose (repère) de la caméra par rapport au repère du monde.

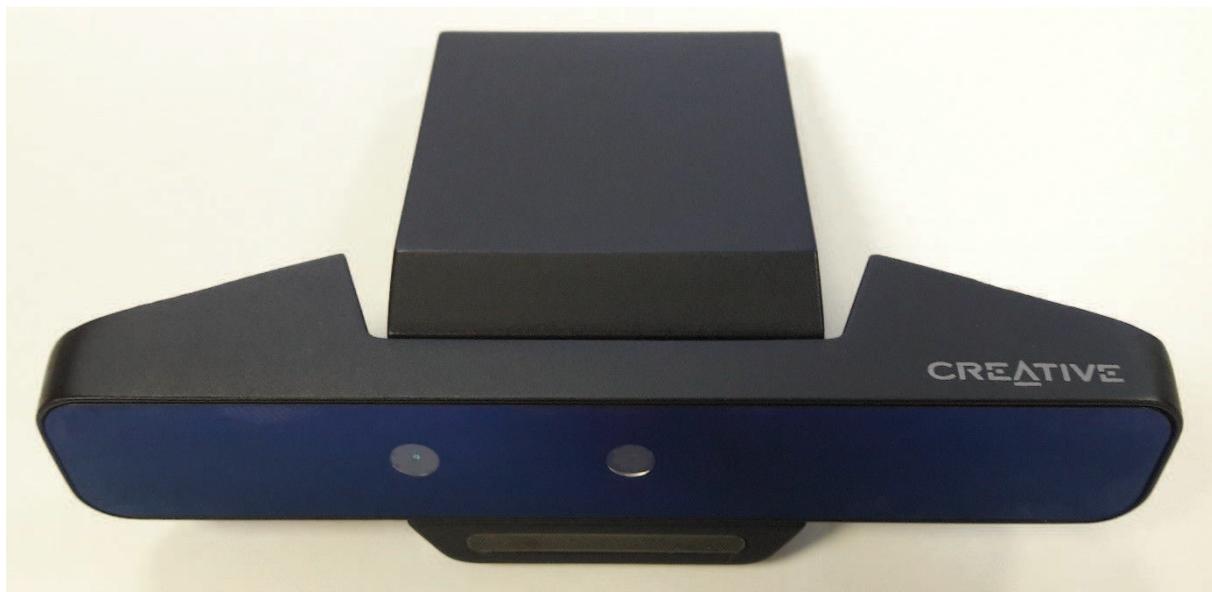


FIGURE 10 – Caméra Intel RealSense SR300 RGB-D. Les capteurs IR (gauche) et RVB (centre) sont clairement visibles sur l'image. Le projecteur IR est caché par la face avant teintée du boîtier.

Un capteur à lumière structurée repose essentiellement sur deux composants électroniques principaux, un projecteur IR et une caméra IR. Ce couple associé à un circuit intégré dédié permet de calculer la profondeur des objets visibles du point de vue de la caméra et qui se trouvent dans son champs d'opérations. Souvent, un capteur RVB fait partie du système pour récupérer l'information de la couleur. Le capteur Occipital Structure [urls] est un exemple de caméra à lumière structurée qui ne dispose pas de caméra RVB.

**1.1.6.2 Paramètres intrinsèques** Le modèle sténopé décrit la projection d'un point du monde 3D dans le plan image (2D) de la caméra.

Considérons un point  $M_c = [x_c, y_c, z_c]^T$  dans le repère de la caméra. Nous voulons exprimer la projection de  $M_c$  dans les coordonnées de l'image que nous notons  $P_c = [u_c, v_c]^T$  en utilisant le modèle sténopé.

Premièrement, nous commençons par normaliser le point

$$M_n = [x_n, y_n]^T = [x_c/z_c, y_c/z_c]^T \quad (17)$$

Dans le modèle de caméra sténopé, les rayons sont considérés comme passant linéairement à travers le centre optique, ce qui n'est pas vrai dans le cas des caméras réelles. En effet, l'utilisation de lentilles modifie la linéarité des rayons lumineux, ce qui entraîne une distorsion non linéaire sur les images finales [Hei00].

En utilisant le point normalisé, la distorsion est appliquée en deux étapes [HKH<sup>+</sup>12] :

$$M_g = [(2k_3x_ny_n + k_4(r^2 + 2x_n^2), k_3(r^2 + 2y_n^2) + 2k_4x_ny_n)] \quad (18)$$

$$M_k = (1 + k_1r^2 + k_2r^4 + k_5r^6)M_n + M_g \quad (19)$$

Où  $r^2 = x_n^2 + y_n^2$  et  $k_c = [k_1, \dots, k_5]$  est le vecteur des coefficients de distorsion.

Le point  $P_c$  que nous recherchons est :

$$\begin{bmatrix} u_c \\ v_c \end{bmatrix} = \begin{bmatrix} f_{cx} & 0 \\ 0 & f_{cy} \end{bmatrix} \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} u_{0c} \\ v_{0c} \end{bmatrix} \quad (20)$$

Les paramètres  $f_{cx}, f_{cy}, p_{0c}, k_1, k_2, k_3, k_4, k_5$  sont appelés les paramètres intrinsèques de la caméra où  $[f_{cx}, f_{cy}]$  sont les distances focales et  $p_{0c} = [u_{0c}, v_{0c}]$  est le point principal. L'étalonnage intrinsèque consiste à trouver ces paramètres. Pour ce faire, la correspondance doit être établie entre un ensemble de points 3D et leurs projections dans le plan image 2D [Sem16].

Zhang [DRS10] a fait la classification suivante pour les techniques d'étalonnage, basée sur la dimensionnalité de la pièce étalon :

- Étalonnage basé sur une pièce étalon 3D : la pièce étalon 3D typique est composé de deux ou trois plans orthogonaux [Hei00]. La géométrie de la pièce doit être connue avec une grande précision.

- Étalonnage 2D : consiste à utiliser un objet planaire tel qu'un damier ou un plan avec des motifs circulaires capturés à partir de différents points de vue. De nombreuses ressources sont disponibles sur le sujet [Zha00], [SM99] ainsi que des outils logiciels [urlc]. La méthode est également implémentée dans la bibliothèque OpenCV [urld].

- Étalonnage à base de ligne 1D : proposé par Zhang [Zha04], il consiste à observer un ensemble de points colinéaires tournant autour d'un point fixe.

- Auto-étalonnage : ou étalonnage 0D comme appelé par Zhang [DRS10] où aucune pièce étalon n'est requise. La méthode consiste à étalonner la caméra à partir d'une séquence d'images d'une scène statique, sans aucune connaissance préalable du mouvement de la caméra [HZ05].

**1.1.6.3 Méthodes d'étalonnage** Bien que construites autour du capteur Kinect v1, la plupart des méthodes que nous citons sont censées être compatibles avec une large gamme de caméras à lumière structurée grand public.

Si la pièce étalon doit être bien connue (forme, couleur, taille ...), la méthode d'étalonnage est dite supervisée. Sinon, la méthode est dite non supervisée.

**1.1.6.3.1 La méthode de Smisek** Smisek et al [SJP13] ont travaillé avec les images RVB et IR d'un damier provenant d'une Kinect v1. Ainsi, ils ont estimé les paramètres intrinsèques des caméras RVB et IR de cette Kinect ainsi que leur pose relative. Les paramètres intrinsèques de la caméra IR représentent également la caméra de "profondeur" de la Kinect qui renvoie une valeur de disparité au lieu d'une valeur de profondeur. La relation entre la disparité et la profondeur est la suivante :

$$d = 1/(c_1z) - c_0/c_1 \quad (21)$$

Où  $c_0$  et  $c_1$  sont des paramètres intrinsèques propres à la Kinect.

Considérons le système de coordonnées de la caméra IR comme repère principal de la Kinect. Notons  $K_{IR}$  la matrice d'étalonnage de la caméra IR,  $k_{IR}$  ses coefficients de distorsion,  $dis$  la fonction de distorsion et  $(u_0, v_0)$  le décalage de pixels IR / profondeur de la Kinect. Alors, un point 3D  $X_{IR}$  peut être écrit en fonction de son correspondant dans l'image de profondeur  $X_D = [x, y, d]$  comme suit :

$$X_{IR} = \frac{1}{c_1d + c_0} dis^{-1} \left( K_{IR}^{-1} \begin{bmatrix} x + u_0 \\ y + v_0 \\ 1 \end{bmatrix}, k_{IR} \right) \quad (22)$$

En utilisant les positions des points d'étalonnage (coins en damier) dans les images de profondeur,  $c_0$  et  $c_1$  ont été optimisés pour satisfaire l'équation ci-dessus. Les auteurs ont remarqué des erreurs résiduelles bien que la Kinect ait été bien calibrée. Ils ont amélioré leurs résultats en calculant une image de correction sur  $z$  et l'ont soustraite de la composante  $z$  de  $X_{IR}$ .

**1.1.6.3.2 La méthode de Herrera** Herrera et al [HKH<sup>+</sup>12] ont utilisé une caméra couleur haute résolution qu'ils ont fixée de façon rigide à une Kinect, afin d'améliorer l'étalonnage global du système (caméra haute résolution/Kinect). La pièce étalon est un plan sur lequel un damier est imprimé ou collé. Pour la caméra externe et la caméra couleur de la Kinect, le damier est utilisé pour obtenir les paramètres intrinsèques ainsi que la pose relative entre les deux caméras. Comme le damier n'est pas visible dans l'image de profondeur, ce sont les quatre coins du plan étalon qui sont sélectionnés, par

un procédé manuel, pour obtenir une estimation initiale de la distance focale, du point principal et des distorsions de la caméra de profondeur. De plus, la caméra de profondeur (de la Kinect) possède cinq paramètres intrinsèques supplémentaires :

- Les deux premiers paramètres  $c_0$  et  $c_1$  que nous avons déjà mentionnés.
- Les trois autres paramètres, à savoir  $D_\delta$ ,  $\alpha_0$  et  $\alpha_1$ , proviennent du modèle de distorsion proposé par les auteurs pour le calcul de la disparité dans la Kinect v1 :

$$d_k = d + D_\delta(u, v).exp(\alpha_0 - \alpha_1 d) \quad (23)$$

Où  $D_\delta$  représente la distorsion spatiale.

Les différents paramètres sont estimés via un processus d'optimisation en trois étapes.

**1.1.6.3.3 La méthode de Bingwen** Bingwen et al [JLG14] ont effectué un étalonnage intrinsèque d'une caméra Kinect v1. Au lieu de travailler avec un damier, les auteurs ont utilisé un ensemble de cuboïdes comme pièce étalon. Les cuboïdes doivent être bien fabriqués avec des dimensions connues, car les dimensions ainsi que les angles entre les surfaces voisines des cuboïdes agissent comme valeurs de référence pour le processus d'étalonnage. Le modèle sténopé, similaire à [HKH<sup>+</sup>12], est utilisé pour l'étalonnage de la caméra IR. Pour la mesure de profondeur, l'équation (21) est utilisée. Bingwen et al ont choisi la décroissance exponentielle (23) introduite par Herrera et al [HKH<sup>+</sup>12] pour la correction de la distorsion de disparité. La fonction objectif est définie comme une combinaison linéaire des erreurs de distance et d'angle :

$$E = S \sum_{view} (\tau E_a + (1 - \tau) E_d) \quad (24)$$

Où  $E_a$  et  $E_d$  représentent respectivement les erreurs sur les angles et les distances, voir [JLG14] pour plus de détails,  $\tau$  est le poids de  $E_a$  et  $S = 1 / \sum_{view} \sum_{i,j} 1$ .

$E_a$  et  $E_d$  sont ensuite réécrits en fonction des paramètres de l'étalonnage. L'algorithme d'optimisation utilisé est celui de Levenberg-Marquardt, qui est appliqué deux fois. Dans la première étape, les paramètres de correction de la distorsion de disparité de l'équation (23) ( $D_\delta, \alpha_0, \alpha_1$ ) sont mis à zéro et les autres paramètres sont optimisés. Alors que dans la deuxième étape, la sortie de l'opération précédente est utilisée comme entrée et cette fois les paramètres de disparité sont optimisés.

**1.1.6.3.4 La méthode de Staranowicz** La méthode de Staranowicz et al [SBMM15] est un algorithme d'étalonnage qui s'adapte à n'importe quelle combinaison de capteurs de couleur et de profondeur. L'algorithme prend en entrée une vidéo d'un objet sphérique en mouvement devant la caméra. Un processus de détection robuste trouve

les ellipses représentant les projections de l'objet sphérique sur les images couleurs ainsi que celles de profondeurs. A partir des paires d'ellipses (couleur/profondeur), l'algorithme déduit la profondeur ainsi que les paramètres intrinsèques de la caméra couleur avec leurs distorsions de lentilles respectives et la position relative du système. Enfin, une étape de minimisation non linéaire est appliquée dans le but de raffiner les paramètres obtenus dans l'étape précédente. Elle consiste en une estimation du maximum de vraisemblance pour tous les paramètres d'étalonnage. La fonction objectif est de la forme suivante :

$$\max_{K_R, K_D, R_{R \rightarrow D}, t_{R \rightarrow D}, k_R, k_D} \rho_1 L_1 + \rho_2 L_2 \quad (25)$$

Où  $L_1$  et  $L_2$  sont des fonctions de log-vraisemblance pondérées par  $\rho_1$  et  $\rho_2$ ,  $K_R$  et  $K_D$  sont les matrices des caméras couleur et IR respectivement,  $k_R$  et  $k_D$  sont les paramètres de distorsion des lentilles et  $R \rightarrow D$  et  $tR \rightarrow D$  définissent la position relative du système . Plus de détails sont donnés dans [SBMM15] pour savoir comment  $L_1$  et  $L_2$  sont formulés.

**1.1.6.3.5 La méthode de Karan** La méthode d'étalonnage de Karan [Kar15] utilise un damier comme objet étalon. Le modèle sténopé a été choisi pour calibrer les capteurs RVB et IR d'une caméra Kinect v1 à l'aide du module Matlab de Bouguet [urlc]. La distorsion tangentielle de chaque capteur a été négligée car les valeurs étaient trop faibles : seule la distorsion radiale a été prise en compte. Pour la profondeur, le modèle de mesure est le suivant :

$$\frac{1}{z} = a_z \frac{1}{z_s} - b_z \quad (26)$$

Où  $z_s$  est la valeur de profondeur calculée par le capteur et  $a_z$  et  $b_z$  sont les paramètres spécifiques du capteur pour la profondeur et ils doivent être déterminés.

Afin d'estimer les paramètres  $a_z$  et  $b_z$ , un ensemble de paires  $(z, z_s)$  est déterminé par la méthode décrite dans [Kar13], et l'ajustement par la méthode des moindres carrés est appliqué au modèle (26).

Cette section constitue une revue générale du pipeline d'acquisition de la géométrie, surtout pour l'étape de l'étalonnage et plus particulièrement l'étalonnage des capteurs à lumière structurée. En effet, nous avons développé une nouvelle méthode d'étalonnage qu'on présentera dans la deuxième partie de ce travail. La section suivante introduira les notions théorique sur l'acquisition de la couleur.

## 2 Acquisition de la couleur

### 2.1 Introduction

Le rendu réaliste d'un objet exige une étude de la réaction de la surface de cet objet vis-à-vis de la lumière. En effet, une équation dite "équation de rendu" qui est basée sur un modèle géré par des paramètres spécifiques permet d'attribuer une couleur à chaque point de l'objet.

L'équation de rendu est une équation intégrale qui exprime la luminance quittant une surface en fonction de la luminance incidente de toutes les directions, pondérée par la fonction de distribution de la réflectance bidirectionnelle BRDF. Ainsi, il s'agit essentiellement de trouver un modèle de BRDF adapté à la surface dont nous voulons générer un rendu.

### 2.2 L'apparence

#### 2.2.1 La radiométrie

Selon la Commission internationale de l'éclairage CIE radiométrie est "la mesure des grandeurs relatives à l'énergie rayonnante" [Com87]. Globalement, la radiométrie s'intéresse à l'ensemble des radiations électromagnétiques. Plus communément, seules les radiations dans la longueur d'onde se situe entre 100 nm et 2500 nm sont prises en compte en radiométrie [urlf].

La radiométrie fait intervenir un ensemble de paramètres : la luminance, le flux, l'éclairement et l'intensité, ainsi que les différentes relations les reliant entre elles.

Pour de telles mesures, des notions géométriques comme l'angle solide, qu'on introduit dans le paragraphe suivant, sont utilisées.

**2.2.1.1 L'angle solide** L'angle solide est le rapport de la projection  $dS$  de la surface élémentaire  $dA$  d'un objet sur une sphère centrée en O et du carré du rayon de cette sphère. Voir la figure 11. L'angle solide est calculé par la formule suivante :

$$d\omega = \frac{dS}{R^2} \quad (27)$$

Unité : stéradian  $sr$

#### 2.2.2 La photométrie

La photométrie est la mesure de l'énergie rayonnante dans la partie visible du spectre électromagnétique perceptible par l'œil humain [urlf]. La CIE définit la photométrie de la manière suivante :

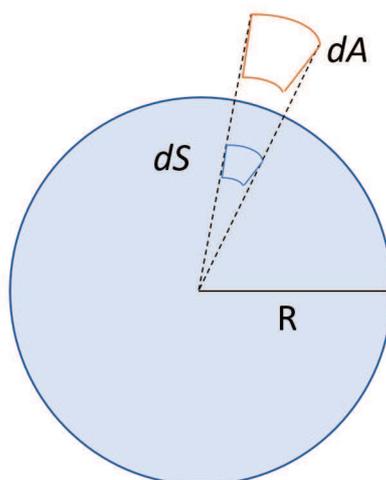


FIGURE 11 – Schéma représentatif de la notion d'angle solide.

C'est « la mesure des grandeurs se rapportant au rayonnement tel qu'il est évalué selon une fonction d'efficacité lumineuse relative spectrale donnée, par exemple  $V(\lambda)$  ou  $V'(\lambda)$  »

Visuellement, "la photométrie dans laquelle l'œil est utilisé pour faire des comparaisons quantitatives entre des stimuli de lumière" [Com87]. Physiquement, "la photométrie dans laquelle des récepteurs physiques sont utilisés pour faire les mesures" [Com87] [urlf].

**2.2.2.1 Le flux énergétique** Le flux énergétique  $\Phi_e$  est la quantité d'énergie rayonnée par une source, par unité de temps.

L'unité du flux énergétique selon le système international est le watt  $W$ .

**2.2.2.2 L'intensité énergétique** L'intensité énergétique ou l'intensité de rayonnement est représentée le flux énergétique par unité d'angle solide.

L'unité d'intensité de rayonnement est le  $Wsr^{-1}$

$$I = \frac{d\Phi_e}{d\omega} \quad (28)$$

**2.2.2.3 La luminance énergétique** La luminance énergétique est le flux  $d\Phi_e$  transmis par un élément de surface  $dA$  et se propageant dans l'élément d'angle solide  $d\omega$ , dans une direction donnée, par unité de surface et par unité d'angle solide projeté. L'unité de la luminance est  $Wm^{-2}sr^{-1}$

$$L(x, w) = \frac{d^2\Phi_e(\omega)}{dA \cdot \cos(\theta) \cdot d\omega} \quad (29)$$

**2.2.2.4 L'éclairement énergétique** L'éclairement énergétique est le rapport entre le flux énergétique reçu  $d\Phi_e$  par une surface et l'aire  $dA$  de cette surface. L'unité de l'éclairement énergétique est  $Wm^{-2}$

$$E = \frac{d\Phi_e}{dA} = \int_{\Omega_i} L_i(x, \omega_i) \cos(\theta_i) d\omega_i \quad (30)$$

**2.2.2.5 La fonction de distribution de réflectance bidirectionnelle** Introduite par Fred E. Nicodemus en 1965 [Nic65], la notion de fonction de distribution de la réflectance bidirectionnelle (BRDF) a pour objectif de fournir une généralisation du calcul de la réflectance d'une surface opaque, indépendamment de la géométrie.

La fonction de distribution de la réflectance bidirectionnelle représente les propriétés géométriques de la réflexion de la lumière dans tout le demi-espace au-dessus de la surface éclairée. C'est le rapport entre la luminance réfléchie dans la direction  $\vec{\omega}_r$  et l'éclairement de la surface provenant de la direction  $\vec{\omega}_l$ .

$$f(x, \vec{\omega}_l \rightarrow \vec{\omega}_r) = \frac{dL_r(x, \vec{\omega}_r)}{dE(\vec{\omega}_l)} \quad (31)$$

Or

$$d\Phi_i(\omega_i) = L_i(x, \omega_i) dA \cdot \cos(\theta_i) \cdot d\omega_i \quad (32)$$

Où  $d\omega_i$  est l'angle solide sous lequel on voit la source depuis  $dA$ , on a donc :

$$f(x, \omega_i \rightarrow \omega_r) = \frac{dL_r(x, \omega_r)}{L_i(x, \omega_i) \cdot \cos\theta_i \cdot d\omega_i} \quad (33)$$

L'unité de la fonction de distribution de la réflectance bidirectionnelle est  $sr^{-1}$ .

### 2.2.3 L'équation de rendu

L'équation de rendu est une équation intégrale qui calcule la luminance quittant une surface en fonction de la luminance incidente de toutes les directions, pondérée par la BRDF [url].

$$L_r(x, \omega_r) = \int_{\Omega} f(x, \omega_i \rightarrow \omega_r) L_i(x, \omega_i) \cos\theta_i d\omega_i \quad (34)$$

### 2.2.4 Les propriétés de la BRDF

Une BRDF est caractérisée par les quatre propriétés suivantes :

- La BRDF est une fonction non bornée, à valeurs positives ou nulles.

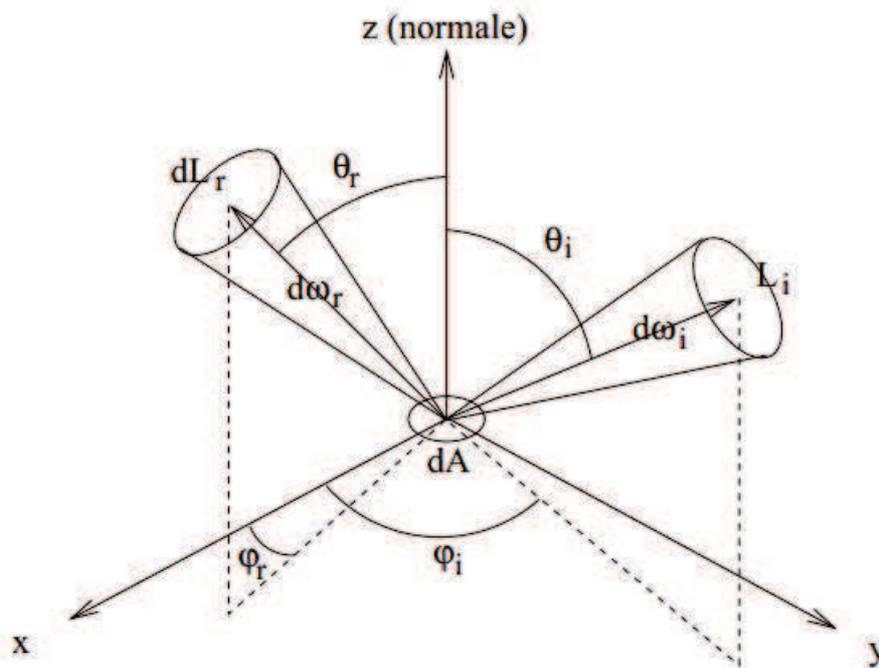


FIGURE 12 – Géométrie de la BRDF [Noe99]

- Réciprocité : La BRDF est une fonction symétrique. Elle satisfait la loi de réciprocité d'Helmholtz [Hel25] qui est aussi appelée loi du retour inverse de la lumière.

$$f_r(x, \omega_i, \omega_r) = f_r(x, \omega_r, \omega_i) \quad (35)$$

- Conservation d'énergie : Le flux réfléchi par une surface élémentaire d'aire  $dA$ , doit être inférieur ou égal au flux incident [Noe99].

$$\frac{\int_{\Omega_r} \int_{\Omega_i} f_r(x, \omega_i, \omega_r) L_i(x, \omega_i) \cos\theta_i \sin\theta_r d\omega_i d\omega_r}{\int_{\Omega_i} L_i(x, \omega_i) \cos\theta_i d\omega_i} < 1 \quad (36)$$

- Anisotropie et isotropie :
  - Anisotropie : La BRDF est anisotrope quand la réflexion de la lumière varie avec la rotation de la surface autour de sa normale.
  - Isotropie : La BRDF est isotrope quand la réflexion est indépendante de la rotation de la surface autour de sa normale.

## 2.3 La géométrie de la réflexion

Considérons une source de lumière, un observateur et une surface. Alors, la géométrie de la réflexion représente les différents vecteurs qui interviennent dans le calcul de la quantité de lumière qui atteint l'observateur. Elle est représentée dans la figure 14.

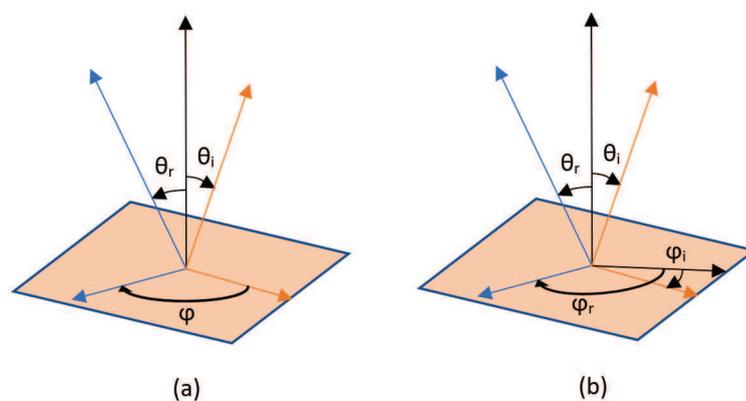


FIGURE 13 – (a) Angles utiles pour une BRDF isotrope. (b) Angles utiles pour une BRDF anisotrope

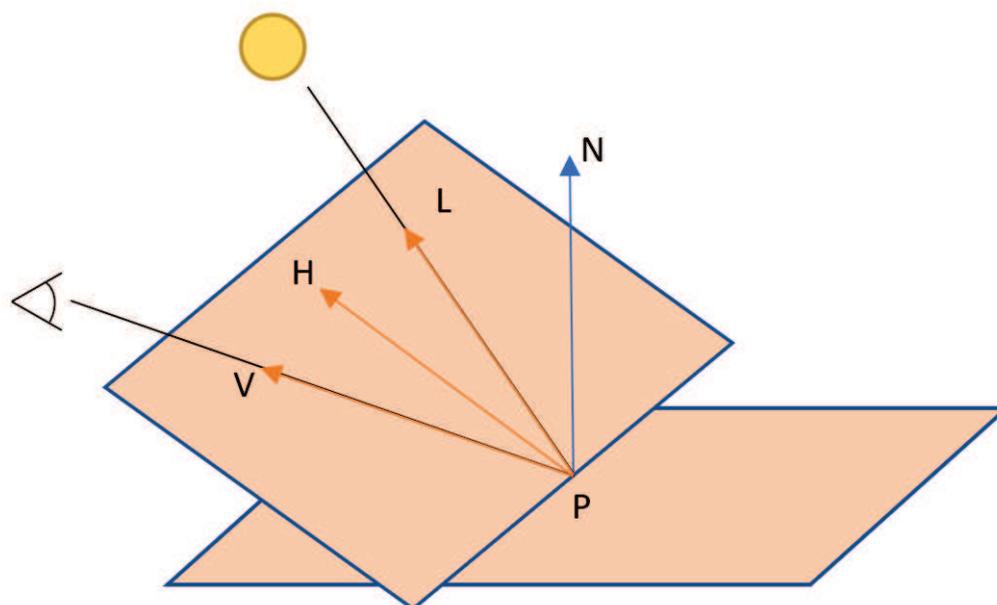


FIGURE 14 – Géométrie de la réflexion : un observateur regarde un point  $P$  sur une surface suivant la direction  $V$ .  $L$  est la direction de la lumière et  $H$  la direction spéculaire

Dans la figure geomReglex, un observateur regarde un point  $P$  sur une surface suivant la direction  $V$ .  $L$  est la direction de la lumière et  $H$  la direction spéculaire qui est définie comme suit :

$$H = \frac{V + L}{\text{length}(V + L)} \quad (37)$$

$\theta$  est l'angle entre  $V$  et  $H$  ainsi qu'entre  $H$  et  $L$ .  $N$  est la normale à la surface et  $\theta$  est l'angle entre  $H$  et  $N$ .

## 2.4 Les modèles de réflexion

### 2.4.1 La réflexion ambiante

La réflexion ambiante représente la lumière provenant uniformément de l'environnement puis réfléctée de façon équitable dans toutes les directions [CT82].

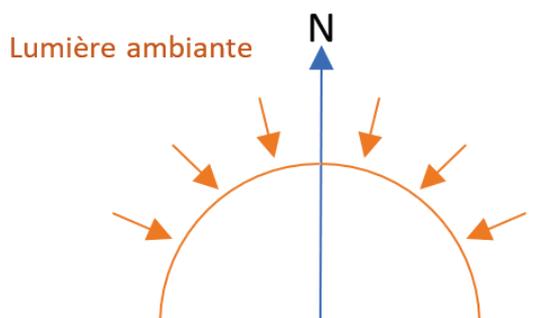


FIGURE 15 – Réflexion ambiante : la lumière provient uniformément de l'environnement puis elle est réfléctée de façon équitable dans toutes les directions.

### 2.4.2 La réflexion spéculaire

Dans le cas d'un miroir lisse parfait, chaque rayon incident est réfléchi par la surface de telle sorte que l'angle entre le rayon incident et la normale à la surface est égal à l'angle entre le rayon réfléchi et la normale. Le rayon incident, le rayon réfléchi et la normale sont coplanaires. C'est la loi de Snell-Descartes pour la réflexion. Voir la figure 16.

Pour un objet spéculaire réel, la lumière est réfléchie dans un cône centré sur la direction définie par les lois de Snell-Descartes. Voir la figure 17.

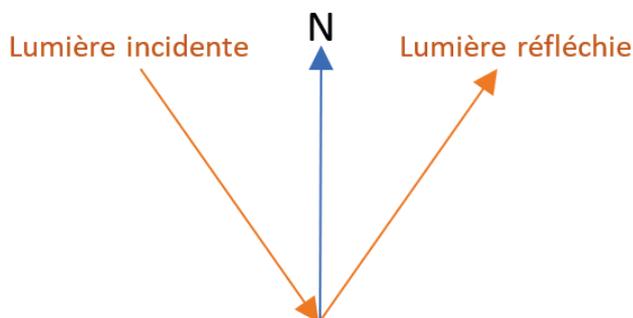


FIGURE 16 – Réflexion spéculaire dans le cas d'un miroir lisse parfait : chaque rayon incident est réfléchi par la surface de telle sorte que l'angle entre le rayon incident et la normale à la surface est égal à l'angle entre le rayon réfléchi et la normale.

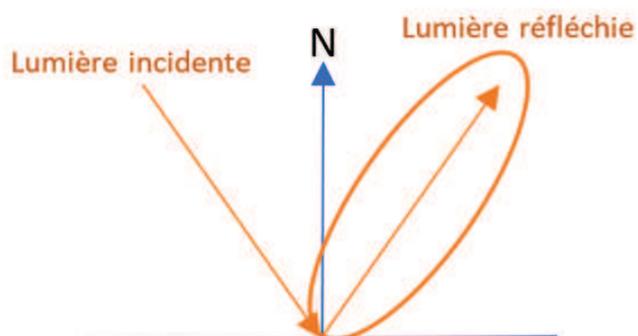


FIGURE 17 – Réflexion spéculaire dans le cas d'un objet spéculaire réel : la lumière est réfléchié dans un cône centré sur la direction définie par les lois de Snell-Descartes.

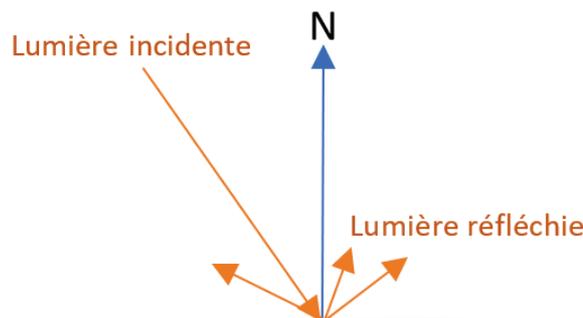


FIGURE 18 – Réflexion diffuse : la lumière est réfléchi dans plusieurs directions.

### 2.4.3 La réflexion diffuse

Il s'agit de réflexion diffuse quand la lumière est réfléchi dans plusieurs directions. C'est un ensemble de réflexions spéculaires sur plusieurs surfaces planes avec des directions aléatoires formant une surface rugueuse. Voir la figure 18.

## 2.5 Quelques modèles de BRDF

Dans ce paragraphes, nous allons présenter quelques modèles de BRDF classiques afin d'illustrer les concepts généraux liés à cette notion. Nous commencerons par deux modèles empiriques : le modèle de Lambert puis le modèle Phong, ensuite nous introduirons les deux modèles analytiques : modèle de Ward et le modèle de Cook-Torrance.

### 2.5.1 Modèles empiriques

**2.5.1.1 Modèle de LAMBERT** Lorsque la lumière est réfléchi dans toutes les directions avec le même rayonnement au lieu qu'elle soit reflétée dans une seule direction, la réflexion est dite Lambertienne. Le modèle de Lambert s'applique aux diffuseurs parfaits [DRS10].

La BRDF de Lambert est définie comme suit :

$$f_{\text{lambert}} = \frac{C}{\pi} \quad (38)$$

L'équation de rendu est :

$$L_r(x, \omega_i, \omega_r) = \frac{C}{\pi} \int_{\Omega_i} L_i(x, \omega_i) \cos\theta_i d\omega_i \quad (39)$$

Avec  $C$  représentant la réflectivité de la surface. Elle varie de 0 pour une surface complètement absorbante à 1 pour une surface réfléchissante.

**2.5.1.2 Modèle de PHONG** Le modèle de Phong est l'un des premiers modèles d'éclairage dans le domaine de la synthèse d'images. Même s'il est physiquement non valide, le modèle de Phong applique un effet tridimensionnel aux objets éclairés par une source lumineuse directionnelle [Pho75].

L'équation de rendu est :

$$L_r(x, \omega_i, \omega_r) = k_d \cdot L_i \cdot (N \cdot \omega_i) + k_s \cdot L_i \cdot (R \cdot \omega_r)^n + k_a \cdot L_a \quad (40)$$

Avec :  $k_d$  la proportion de la réflexion diffuse,  $k_s$  la proportion de la réflexion spéculaire et  $k_a$  la proportion de la réflexion ambiante.

## 2.5.2 Modèles analytiques

**2.5.2.1 Modèle de Ward** Le modèle de Ward diffère du modèle de Phong par l'utilisation de la fonction exponentielle au lieu du cosinus. Le terme exponentiel est contrôlé par la variable de rugosité de surface microscopique  $\alpha$  [War92].

— Le modèle gaussien isotrope :

$$f_r(\omega_i, \omega_r) = \frac{k_d}{\pi} + k_s \frac{1}{\sqrt{\cos\theta_i \cos\theta_r}} \frac{e^{-\tan^2(\frac{N \cdot H}{\alpha^2})}}{4\pi\alpha^2} \quad (41)$$

Avec :  $k_d$  la proportion de la réflexion diffuse,  $k_s$  la proportion de la réflexion spéculaire et  $\theta_H$  l'angle entre la normale et le vecteur  $H$ .

— Le modèle gaussien anisotrope :

$$f_r(\omega_i, \omega_r) = \frac{k_d}{\pi} + k_s \frac{1}{\sqrt{\cos\theta_i \cos\theta_r}} \frac{1}{4\pi\alpha_x\alpha_y} e^{-2 \frac{\frac{H \cdot x^2}{\alpha_x} + \frac{H \cdot y^2}{\alpha_y}}{1 + (H \cdot N)}} \quad (42)$$

Avec :  $x$  un vecteur unitaire sur le plan de la surface (tangente),  $y$  un vecteur unitaire sur le plan de la surface perpendiculaire à  $x$  (bitangente),  $\alpha_x$  et  $\alpha_y$  représentent les variables de rugosité dans les directions  $x$  et  $y$  respectivement.

## 2.5.2.2 Modèle de COOK-TORRANCE

**2.5.2.2.1 Présentation** Le modèle de Cook-Torrance est considéré comme très complet physiquement. Dans ce modèle, chaque surface est considérée comme composée de microfacettes. Ces microfacettes sont supposées être des réflecteurs parfaits et seules celles orientées dans la direction de  $H$  contribuent au calcul de la composante spéculaire du modèle. Le modèle de Cook-Torrance fait appel au modèle de Lambert pour la composante diffuse. La BRDF de Cook-Torrance est définie comme suit [CT82] :

$$f_r = k_d f_{\text{lambert}} + k_s f_{\text{cook-torrance}} \quad (43)$$

Avec  $k_d$  la proportion de la réflexion diffuse et  $k_s$  la proportion de la réflexion spéculaire. On suppose que  $k_d + k_s \leq 1$  afin de garantir la conservation de l'énergie.

$f_{lambert} = C/\pi$  comme vu précédemment. Ainsi, pour trouver la valeur de  $f_r$  il faut calculer le terme  $f_{cook-torrance}$  :

$$f_{cook-torrance} = \frac{D.F.G}{4(N.L).(N.V)} \quad (44)$$

Avec  $D$  la fonction de distribution de microfacettes,  $F$  la fonction de Fresnel et  $G$  la fonction d'atténuation géométrique.

Nous présenterons les notions de fonction de distribution de microfacettes, fonction de Fresnel et fonction d'atténuation géométrique dans les trois paragraphes qui suivent.

**2.5.2.2 La fonction de distribution des microfacettes** La fonction de distribution des microfacettes est la fraction des facettes qui sont orientées dans la direction  $H$ . Il existe plusieurs fonctions de distribution. Elle est notée  $D$ . Voici quelques exemples de fonctions de distribution de microfacettes.

**Distribution de Beckmann** : La distribution de Beckmann est très utilisée [CT82], car elle peut représenter un large nombre de matériaux. Elle est calculée de la manière suivante :

$$D_{Beckmann} = \frac{1}{m^2 \cos^4 \alpha} e^{-\left(\frac{\tan \alpha}{m}\right)^2} \quad (45)$$

Où  $m$  est la variable qui contrôle la rugosité de la surface.

**Distribution GGX (Trowbridge-Reitz)** : La distribution GGX (Trowbridge-Reitz) se présente sous la forme suivante [WMLT07] :

$$D_{GGX} = \frac{\alpha^2}{\pi((n.m)^2(\alpha^2 - 1) + 1)^2} \quad (46)$$

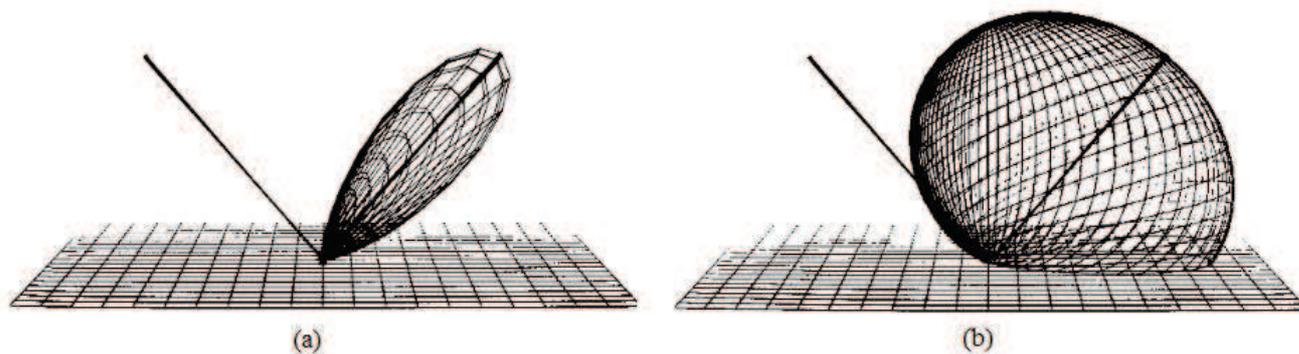
Où  $m$  représente toujours la variable contrôlant la rugosité de la surface.

**Distribution GGX anisotropique** : La distribution GGX anisotropique s'écrit comme suit [Hil] :

$$D_{GGX_{anisotropique}} = \frac{1}{\pi \alpha_x \alpha_y} \frac{1}{\left(\frac{(x.m)^2}{\alpha_x^2} + \frac{(y.m)^2}{\alpha_y^2} + (n.m)^2\right)^2} \quad (47)$$

Où  $m$  représente toujours la variable qui contrôle la rugosité de la surface.  $\alpha_x$  et  $\alpha_y$  représentent les variables de rugosité dans les directions  $x$  et  $y$  respectivement.

Si une surface est composée de plusieurs niveaux de rugosité, alors, la fonction de distribution est égale à la somme pondérée des différentes fonctions de distribution représentant chaque niveau de rugosité [CT82] :


 FIGURE 19 – Distribution de Beckmann (a)  $m = 0.2$  (b)  $m = 0.6$  [CT82].

$$D = \sum_j \omega_j D(m_j) \quad (48)$$

Où, pour une distribution  $j$ ,  $m_j$  représente la variable qui contrôle la rugosité et  $w_j$  est le poids correspondant à cette distribution. La somme des poids est égale à 1.

**2.5.2.2.3 Le coefficient de Fresnel** Le coefficient de Fresnel permet de calculer la quantité de lumière réfléchiée et celle transmise. Parmi les formules de l'estimation du coefficient de Fresnel  $F$ , nous pouvons citer l'équation introduite par Schlick en 1994 [Sch94]. La formule de Schlick est la suivante :

$$F = F_0 + (1 - F_0)(1 - \cos\theta)^5 = F_0 + (1 - F_0)(1 - (H.V))^5 \quad (49)$$

Où :  $F_0 = \left(\frac{\eta_1 - \eta_2}{\eta_1 + \eta_2}\right)^2$

Avec  $\eta_1$  et  $\eta_2$  les indices de réfraction des deux milieux (par exemple, l'air et le matériau de la surface).

**2.5.2.2.4 La fonction d'atténuation géométrique** La fonction d'atténuation géométrique, décrit le comportement de la lumière avec les microfacettes. Elle est notée  $G$  [Bli77].

Le phénomène d'ombrage est causé par le blocage la lumière incidente par des microfacettes avant qu'elle n'atteigne d'autres microfacettes. Voir la figure 20.

$$G_{ombrage} = \frac{2(N.H)(N.L)}{(L.H)} \quad (50)$$

Le phénomène de masquage se produit quand la lumière réfléchiée subie des rebondissements sur quelques microfacettes avant qu'elle n'atteigne l'observateur. Voir la figure 21.

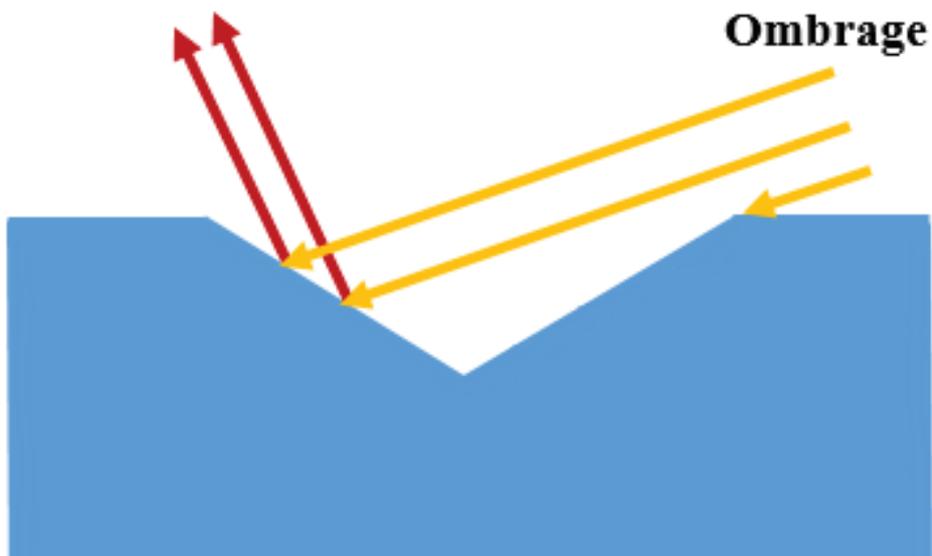


FIGURE 20 – Ombrage de la lumière incidente.

$$G_{\text{masquage}} = \frac{2(N.H)(N.V)}{(V.H)} \quad (51)$$

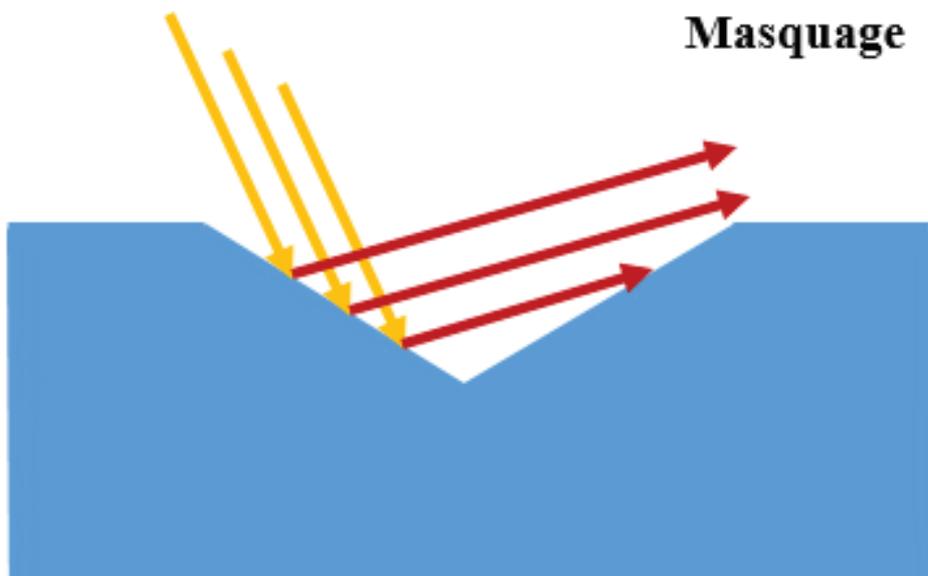


FIGURE 21 – Masquage de la lumière réfléchie.

La formule d'atténuation géométrique est calculée comme suit :

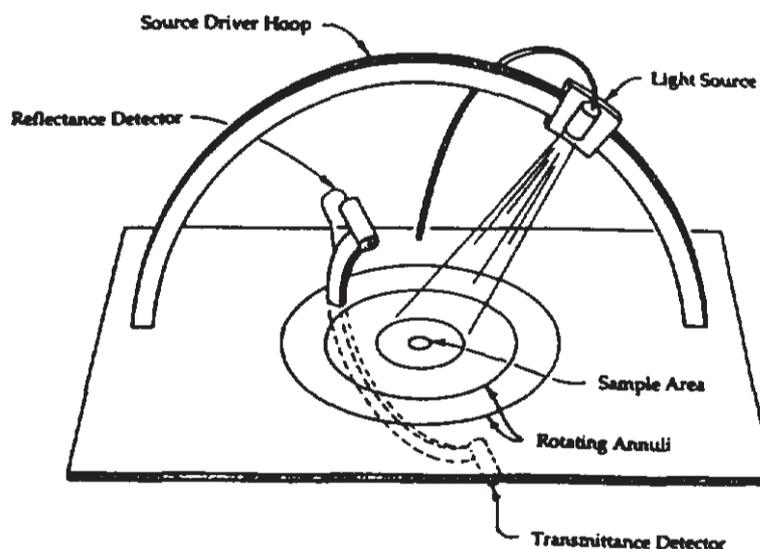
$$G = \min(1, G_{\text{masquage}}, G_{\text{ombrage}}) = \min(1, \frac{2(N.H)(N.V)}{(V.H)}, \frac{2(N.H)(N.L)}{(L.H)}) \quad (52)$$

Avec  $G = 1$  lorsque la lumière ne subit ni masquage ni ombrage.

## 2.6 Dispositif de mesure de BRDF

### 2.6.1 Goniorélectomètre classique

Il s'agit d'un dispositif de mesure de BRDF. Il est composé essentiellement d'un capteur et d'une source lumineuse déplaçable au-dessus de l'échantillon à mesurer, de telle sorte à couvrir les quatre degrés de libertés nécessaires à la mesure de la BRDF.



**Figure 1.** A conventional gonioreflectometer with movable light source and photometer.

FIGURE 22 – Goniorélectomètre classique.[War92]

### 2.6.2 Goniorélectomètre basé sur l'image

Le groupe Lighting Systems Research du laboratoire Lawrence de Berkeley a développé un nouveau goniorélectomètre [Hel25] basé sur l'image, voir la figure 23. Il s'agit d'un dispositif qui utilise un miroir semi-réfléchissant de la forme d'un hémisphère dans le but de capturer d'un seul coup l'intégrité de la lumière réfléchie couverte par le miroir. Ceci offre un avantage par rapport au goniorélectomètre classique en termes de rapidité de la capture de BRDF.

Sur cette partie, nous avons exposé l'état d'art lié à cette thèse. Dans la partie suivante, nous présenterons une nouvelle méthode d'étalonnage pour les capteurs à lumière structurée.

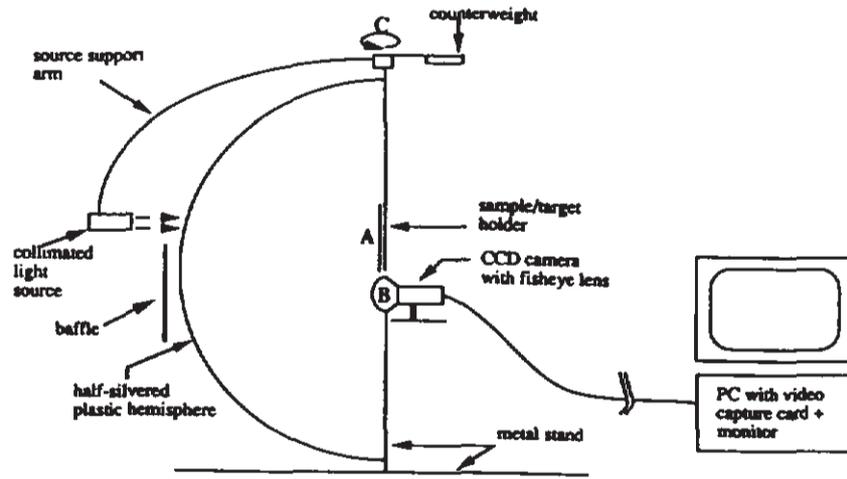


FIGURE 23 – Gonioreflectomètre basé sur l'image.[War92]

## Deuxième partie

### Étalonnage

## 1 Introduction

Ce chapitre décrit une nouvelle méthode d'étalonnage pour l'Intel RealSense SR300 [urlj] avec une double réalisation :

- Amélioration de la précision par rapport à l'étalonnage du fabricant ;
- Proposition d'une méthode d'étalonnage à usage général pouvant être appliquée à des dispositifs similaires au capteur RealSense SR300 ;

Notre algorithme consiste en deux étapes principales :

- Un étalonnage classique par damier qui est un étalonnage 2D pour corriger les rayons de la caméra (caméra IR).

- Une correction de la profondeur calculée une seule fois par capteur à l'aide d'une machine à mesure tri-dimensionnelle (MMT) pour des mesures de référence de haute précision .

La sortie de l'algorithme est un fichier de données d'étalonnage avec les paramètres de la caméra et une grille 3D de coefficients de correction couvrant le domaine d'étalonnage dans le champ de vue de la caméra de profondeur.

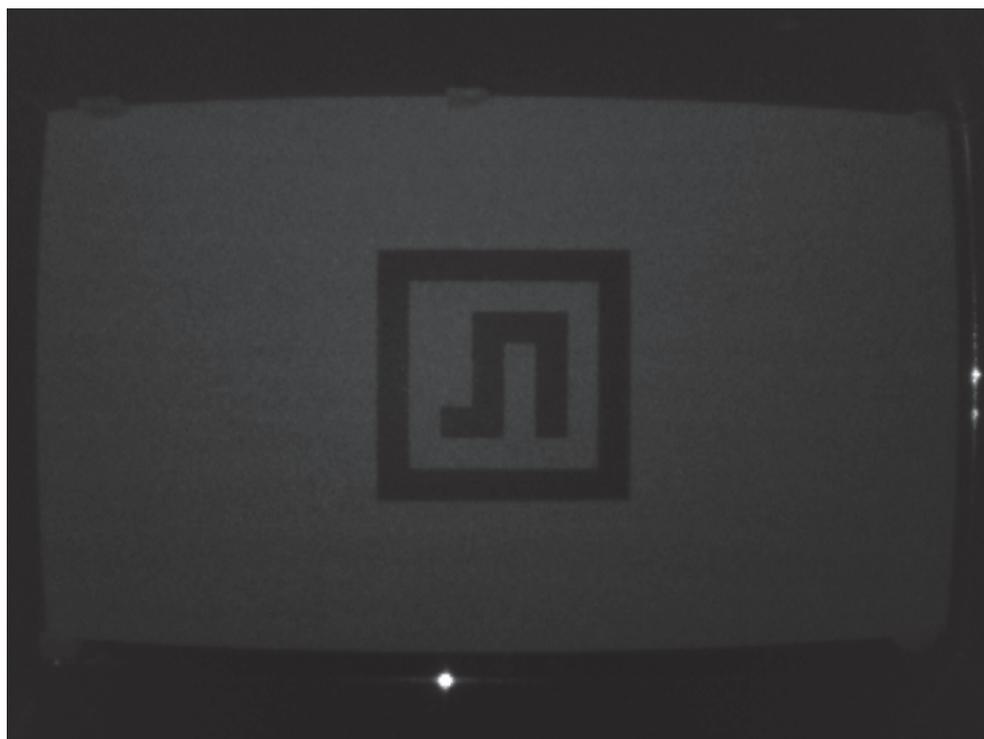


FIGURE 24 – Distorsion dans le SR300 IR. Le panneau avec le motif est rectangulaire.

## 2 Étalonnage 2D

Comme indiqué précédemment, pour obtenir des paramètres intrinsèques plus précis de la caméra (c'est-à-dire afin d'éliminer la distorsion représentée sur la figure 24), nous utilisons un étalonnage classique en damier. Nous photographions un damier à partir de différents points de vue en utilisant l'appareil photo, et utilisons simplement le module d'étalonnage OpenCV pour calculer les paramètres de la caméra. Dans notre cas, nous sommes intéressés par les valeurs intrinsèques du capteur IR.

En pratique, nous utilisons les valeurs intrinsèques pour calculer les coordonnées du point. La relation entre un point 3D  $(x, y, z)$  dans l'espace et son correspondant  $(u, v)$  dans l'image de profondeur est la suivante :

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{(u-p_x)z}{f_x} \\ \frac{(v-p_y)z}{f_y} \end{pmatrix} \quad (53)$$

Où :  $(f_x, f_y)$  est la distance focale et  $(p_x, p_y)$  les coordonnées du centre optique. La coordonnée  $z$  est la profondeur que le capteur renvoie pour le pixel de l'image en profondeur  $(u, v)$ .

Enfin, nous appliquons sur  $x$  et  $y$  un schéma de compensation de distorsion itératif similaire à celui utilisé dans OpenCV.

La correction sur les axes  $X$  et  $Y$  équivaut à corriger les directions des rayons de la caméra.

Maintenant, nous devons ajuster la position de chaque point acquis tout le long de son rayon de caméra correspondant.

## 3 Étalonnage de la profondeur

À cette étape, nous calculons une grille 3D régulière de coefficients de correction sur la totalité ou une partie du champ d'opération du capteur (une pyramide tronquée). Un ensemble de captures d'un plan d'étalonnage est utilisé pour "alimenter" les nœuds de la grille en termes de coefficients de correction. Le processus consiste en deux étapes principales :

- Acquisition de données : points "réels" répartis sur le domaine d'étalonnage et leurs corrections.

- Définition de la grille et remplissage des nœuds : points «virtuels» intégrant les informations de correction locales et répartis régulièrement sur le domaine d'étalonnage.

Pour un capteur donné, ces étapes ne sont effectuées qu'une seule fois pour définir sa grille de correction.



FIGURE 25 – Le plan d'étalonnage et sa configuration sur le CMM.

### 3.1 Acquisition des données

Les données d'entrée sont un ensemble de points, capturés par la caméra 3D à lumière structurée que nous voulons calibrer, étalés sur le domaine d'étalonnage qui est le sous-espace défini par la grille de correction. Chaque point devrait avoir un coefficient de correction.

Pour ce faire, nous avons utilisé comme objet étalon un panneau rectangulaire blanc mat avec un motif imprimé en son centre. Nous plaçons le panneau contre la paroi interne de la MMT, voir la figure 25. Nous ajustons l'orientation du capteur de sorte qu'il soit parallèle au plan d'étalonnage (plus de détails sur le plan et les réglages du capteur sont donnés dans la section dédiée au setup matériel). Des captures successives du plan étalon sont acquises en partant du point le plus loin dans le domaine d'étalonnage et en déplaçant le capteur vers le plan avec un pas fixe jusqu'à ce que tout le domaine soit couvert.

Le processus d'étalonnage 2D permet de corriger les coordonnées  $X$  et  $Y$ , c'est-à-dire les rayons de la caméra. Par conséquent, pour chaque point acquis (du plan), le coefficient de correction que nous cherchons devrait faire glisser le point en avant ou en arrière le long du rayon de la caméra afin que la profondeur du point s'approche le plus de la profondeur réelle. En d'autres termes, nous recherchons la distance réelle entre le plan et le capteur pour calculer le coefficient de correction. Pour calculer la distance réelle entre le plan et le capteur, nous utilisons le traitement d'image pour détecter le motif imprimé sur le plan de calibrage et nous appliquons le principe des triangles similaires en utilisant la distance focale que nous avons déjà calculée avec la méthode du damier. Une fois la

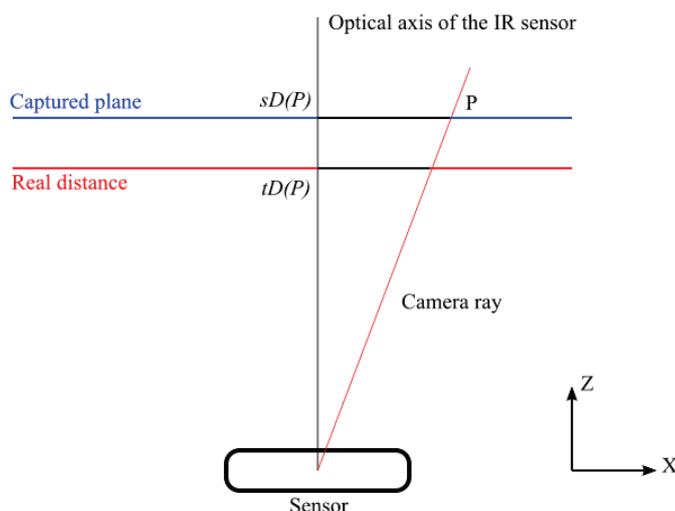


FIGURE 26 – Coefficient de correction pour un point donné  $P$  : la profondeur réelle du plan d'étalonnage  $tD(P)$  divisée par la coordonnée  $z$  de  $P$  renvoyée par le capteur  $sD(P)$ .

première distance trouvée, nous utilisons le pas pré-programmé de la MMT pour déduire les distances suivantes pour les captures successives du plan étalon.

Le coefficient de correction d'un point donné  $P$  est égal à la distance réelle du plan  $tD(P)$ , qui est la profondeur réelle, divisée par la profondeur renvoyée par le capteur  $sD(P)$  comme indiqué sur la figure 26. Par conséquent, le coefficient de correction  $c(P)$  est :

$$c(P) = \frac{tD(P)}{sD(P)} \quad (54)$$

## 3.2 Définition de la grille et remplissage des nœuds

### 3.2.1 Définition de la grille

La grille 3D est un ensemble de nœuds en forme de pyramide tronquée distribués de manière régulière sur le domaine d'étalonnage. Chaque nœud est un vecteur 4D tel que les trois premières composantes sont les coordonnées  $(x, y, z)$  du nœud et la quatrième composante représente le coefficient de correction correspondant au nœud. Les nœuds ne sont pas réellement des points acquis par le capteur, mais plutôt des points «virtuels» intégrant les informations de correction de leur voisinage.

La forme de la grille a été choisie afin de garantir une répartition équitable des points contribuant au calcul de la correction dans chaque nœud, quelle que soit la distance par rapport au capteur.

Nous divisons l'axe  $Z$  en fonction avec un pas fixe. Nous utilisons le même pas pour

les captures progressives du plan d'étalonnage, c'est-à-dire e pas de la MMT.

Pour les axes  $X$  et  $Y$ , nous utilisons également des pas fixes. De plus, nous prenons en compte la résolution maximale du capteur de profondeur que nous ne devons pas dépasser dans notre division de l'espace d'étalonnage.

Enfin, il est aussi important de considérer le nombre approximatif de points qui contribueront à la correction d'un nœud par interpolation. Ce nombre dépend directement de la résolution de notre grille 3D.

### 3.2.2 Remplissage des nœuds

Les positions des nœuds sont définies par la construction de la grille. Cependant, nous devons calculer la correction d'erreur dans chaque nœud. Pour ce faire, nous commençons par définir le voisinage d'un nœud. Ainsi, on définit le voisinage d'un nœud comme toutes les cellules auxquelles il appartient. En utilisant les points des captures du plan d'étalonnage, nous interpolons chaque sous-ensemble de points appartenant à un voisinage afin de calculer la correction de son nœud correspondant. En effet, chaque nœud intègre les informations de correction du sous-espace défini par son voisinage.

Pour interpoler sur les voisinages définis, nous avons opté pour la méthode d'interpolation dite pondération inverse à la distance. Elle est définie comme suit : Soit  $P$  le point à corriger (le nœud),  $P_i, i = 1..N$  les sommets de son voisinage,  $d(P, P_i)$  la distance entre le nœud  $P$  et le voisin  $P_i$ ,  $c_i$  le coefficient de correction du voisin  $P_i$ ,  $p$  un coefficient de lissage et  $c(P)$  le coefficient de correction du nœud, celui que nous recherchons.

$$\omega_i(P) = \frac{1}{d(P, P_i)^p} \quad (55)$$

Le coefficient de lissage  $p$  permet de contrôler l'influence des voisins qui sont loin. On a pris  $p = 3$  dans notre cas.

Une fois remplie, la grille peut être utilisée pour corriger les points de n'importe quel nuage capturé avec le capteur étalonné. La seule condition est que le point appartienne au domaine de calibration préalablement défini.

## 4 Application de la correction

Pour pouvoir être corrigé, un nuage de points capturé doit appartenir partiellement ou totalement au domaine défini par la grille de correction. C'est-à-dire que tout point situé en dehors de la zone d'étalonnage ne peut pas être rectifié.

Soit  $PC$  un nuage de points capturé avec un capteur calibré et  $G$  sa grille de correction. Pour tout point  $P$  appartenant à  $PC$ , nous commençons par trouver la cellule englobante  $BC$  du point dans la grille  $G$ . De telle sorte que l'interpolation par pondération inverse à

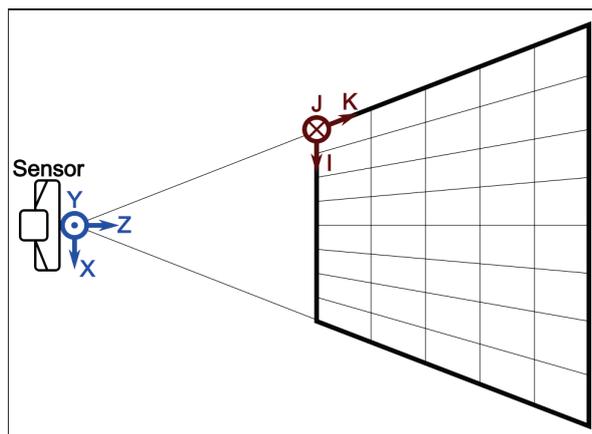


FIGURE 27 – Vue de dessus de l'espace 3D nouvellement défini,  $IJK$  (vue de dessus).

la distance peut être appliquée aux nœuds de la cellule  $BC$  afin de calculer le coefficient de correction pour le point  $P$ . Au final, nous multiplions  $P$  par le coefficient calculé pour le rectifier.

Pour déterminer la cellule englobante d'un point donné, nous définissons une grille 3D (une pyramide tronquée) dans laquelle les cellules sont numérotées suivant  $IJK$  (la direction  $K$  suit chaque rayon du centre de la caméra de notre domaine de calibration). Les coordonnées  $(i, j, k)$  se réfèrent à la cellule avec le sommet en haut à gauche-avant (du point de vue du capteur.) Voir la figure 27.

Par conséquent, en plus des coordonnées  $(x, y, z)$  d'un point donné  $M(x, y, z)$ , nous avons simplement défini de nouvelles coordonnées  $(i, j, k)$  dans la grille  $IJK$  qui indiquent la cellule englobante du point. Nous avons procédé comme suit :

1 - Nous commençons par trouver la coordonnée  $K$ . En effet, pour un  $k$  donné, tous les nœuds correspondant au «niveau»  $k$  partagent la même profondeur. Ainsi, pour chaque niveau, nous pouvons comparer la profondeur du point actuel au premier nœud de chaque niveau, en partant du niveau le plus éloigné au capteur. Le premier niveau pour lequel la profondeur du premier nœud est inférieure à la profondeur du point en question définit la composante  $K$ . Ainsi, la cellule englobante que nous recherchons est à ce niveau.

2 - Pour trouver la coordonnée  $J$ , nous limitons notre recherche au  $k^{\text{ième}}$  niveau obtenu à l'étape précédente. Nous calculons un angle signé entre  $OM_{YZ}$  et l'axe  $Z$ , où  $OM_{YZ}(0, y, z)$  est la projection orthogonale de  $M$  sur le plan  $YZ$ . Nous comparons cet angle aux angles signés calculés entre les projections sur le plan  $YZ$  du premier nœud de chaque rangée à partir du niveau  $k$ , et de l'axe  $Z$ .

3 - Pour la coordonnée  $I$ , nous limitons notre recherche au  $k^{\text{ième}}$  niveau obtenu à partir de la première étape, et à la  $j^{\text{ième}}$  ligne obtenue à partir de la deuxième étape. Nous calculons un angle signé entre  $OM_{XZ}$  et l'axe  $Z$ , où  $OM_{XZ}(x, 0, z)$  est la projection orthogonale de  $M$  sur le plan  $XZ$ . Nous comparons cet angle aux angles signés calculés

entre les projections sur le plan  $XZ$  de chaque nœud de la  $j^{\text{ième}}$  rangée à partir du  $k^{\text{ième}}$  niveau, et l'axe  $Z$ .

## 5 Setup matériel

### 5.1 1er prototype

Avant de penser à utiliser la MMT, nous avons construit un dispositif mécanique. Il s'agit d'un axe encodeur, voir la figure 28. Le capteur est déplacé le long de son axe via une courroie mue par un moteur pas-à-pas qui est lui même piloté par une carte arduino. Voir le schéma de la figure 29.



FIGURE 28 – Notre 1er prototype.

Le problème sur ce système était l'instabilité mécanique lors de chaque arrêt du capteur. Ceci nous a poussé à choisir la MMT comme dispositif pour déplacer le capteur devant le plan étalon.

### 5.2 Setup final

Nous fixons le plan étalon contre le panneau intérieur de la MMT en utilisant de la pâte à modeler. En fait, ceci permet d'ajuster le plan de telle sorte qu'il soit orthogonal

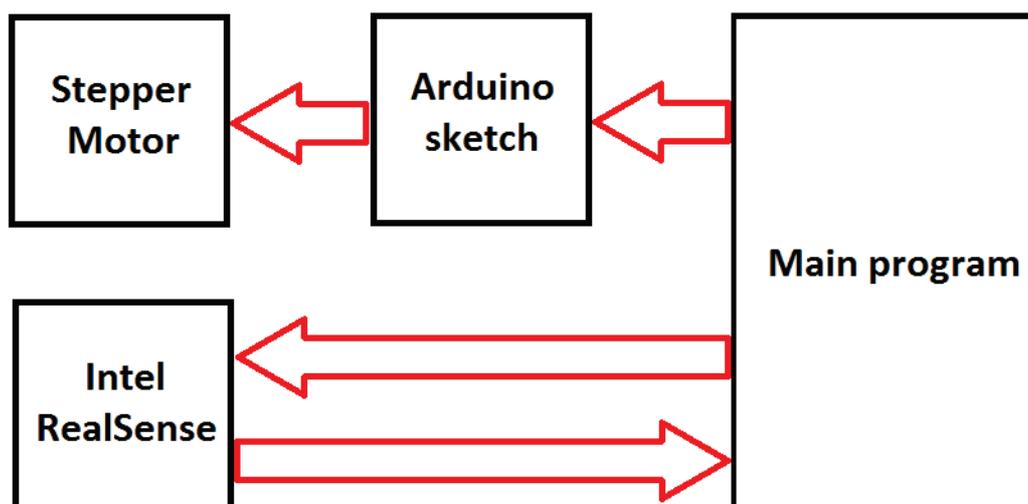


FIGURE 29 – Schéma de fonctionnement du prototype.

à l'axe  $Y$  de la MMT. Nous attachons un palpeur mécanique sur la tête de la MMT et nous "dessinons" un rectangle près de la bordure du plan étalon. Le palpeur est supposé toucher le plan étalon pendant tout le parcours de la trajectoire. Si le test échoue dans une zone du plan, nous compensons le déplacement du plan de calibrage avec la pâte à modeler. La figure 30 montre notre configuration.

Une fois que le plan de calibrage est correctement réglé, nous détachons le palpeur mécanique de la MMT et nous le remplaçons par une rotule pour trépied à ajustement fin. C'est sur la rotule que nous attachons notre capteur à étalonner. Ensuite, nous traquons le marqueur sur le plan étalon et utilisons la rotule pour affiner l'orientation du capteur. Pour ce faire, nous effectuons la détection sur le flux de la caméra IR et nous mettons en évidence les coins du marqueur lorsqu'ils s'alignent sur l'axe  $X$  ou l'axe  $Y$  du capteur. Nous alignons les coins deux à deux, par exemple en le coin haut/gauche avec le coin haut/droite puis le coin haut/gauche avec le coin bas/gauche. C'est-à-dire que nous effectuons l'alignement suivant une direction à la fois (figure 31).

Lorsque les quatre coins du marqueur s'alignent, ce qui signifie que le capteur est parallèle au plan d'étalonnage, nous utilisons le joystick de la MMT pour déplacer le capteur sur les axes  $XY$  de la MMT afin que le centre du marqueur corresponde au centre optique du capteur dans l'image IR. Nous rappelons que le centre optique a été calculé lors de l'étalonnage du damier. Par conséquent, nous pouvons appliquer le principe des triangles similaires pour calculer notre distance référence.

Afin d'améliorer la détection du marqueur, nous éteignons le projecteur IR du capteur et utilisons une source de lumière infrarouge externe pour éclairer le plan afin d'obtenir un éclairage IR continu. En effet, le projecteur IR du capteur projette des bandes pour le

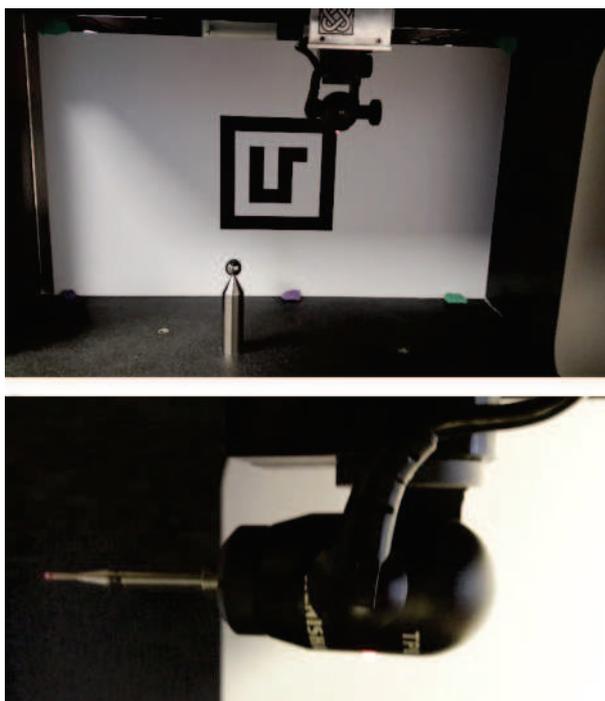


FIGURE 30 – Dessus : le plan d'étalonnage se trouvant sur le "panneau intérieur" de la MMT. En bas : la sonde mécanique utilisée pour vérifier l'orthogonalité du plan avec l'axe Y de la MMT.

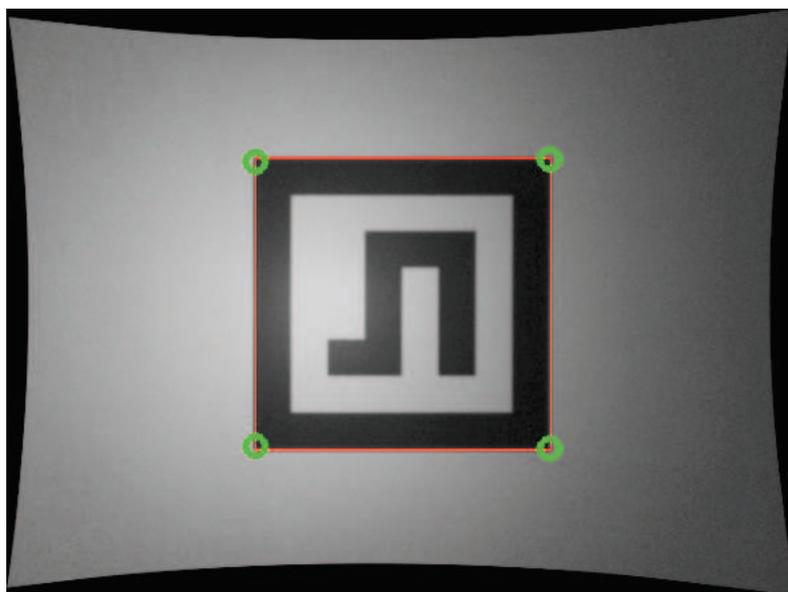


FIGURE 31 – Un alignement réel réussi ; nous avons utilisé des cercles verts super-posés sur le flux IR pour mettre en évidence les coins alignés.

calcul de disparité.

Une fois la distance mesurée, on pulvérise une poudre mate blanche pour masquer le marqueur afin d'éviter que la couleur noire du marqueur altèrent les points qui lui correspondent dans le nuage de point. En fait, le capteur Intel RealSense SR300 a du mal à capturer les surfaces de couleurs noires d'après notre propre expérience.

Pour pouvoir être rectifiable, un nuage de points capturé doit appartenir partiellement ou totalement au domaine défini par la grille de correction. C'est-à-dire que tout point situé en dehors de la zone d'étalonnage ne peut pas être corrigé.

## 6 Résultats et validation de la méthode

### 6.1 Domaine d'étalonnage

Selon les dimensions internes de l'espace de travail de la MMT, et pour que le plan étalon couvre entièrement le «cadre» de chaque nuage de points capturé, nous avons défini notre domaine d'étalonnage comme le sous-espace du champ de vue de la caméra de profondeur entre 10 cm et 27 cm environ du centre de la caméra IR. La grille de correction est de taille 64x48x50.

### 6.2 Étalonnage 2D

Nous avons effectué un étalonnage par damier sur le capteur IR donnant les résultats sur le tableau 1. Nous avons pris 48 photos d'un damier en utilisant une résolution de 640x480. Le damier a 10x8 carrés de 3 cm de bord.

La figure 32 montre une image du damier avant et après la correction via les valeurs de distorsion calculées. Voir le tableau 2 pour les résultats numériques.

TABLE 1 – Les valeurs de l'étalonnage par damier comparées aux valeurs du SDK.

Paramètre	Nos valeurs	Valeurs Intel (extraites du SDK)
Distances focales (pixels)	(473.448, 473.073)	(474.263, 474.263)
Point principal (pixels)	(308.148, 242.341)	(304.816, 245.449)
Distorsion radiale	(-0.117456, -0.0642003, 0.0390934)	(-0.120845, -0.0660312, 0.0516015)
Distorsion tangentielle	(-0.00148510, 0.000892128)	(-0.00265185, -0.00182552)
Erreur moyenne de reprojection	0.64	4.79

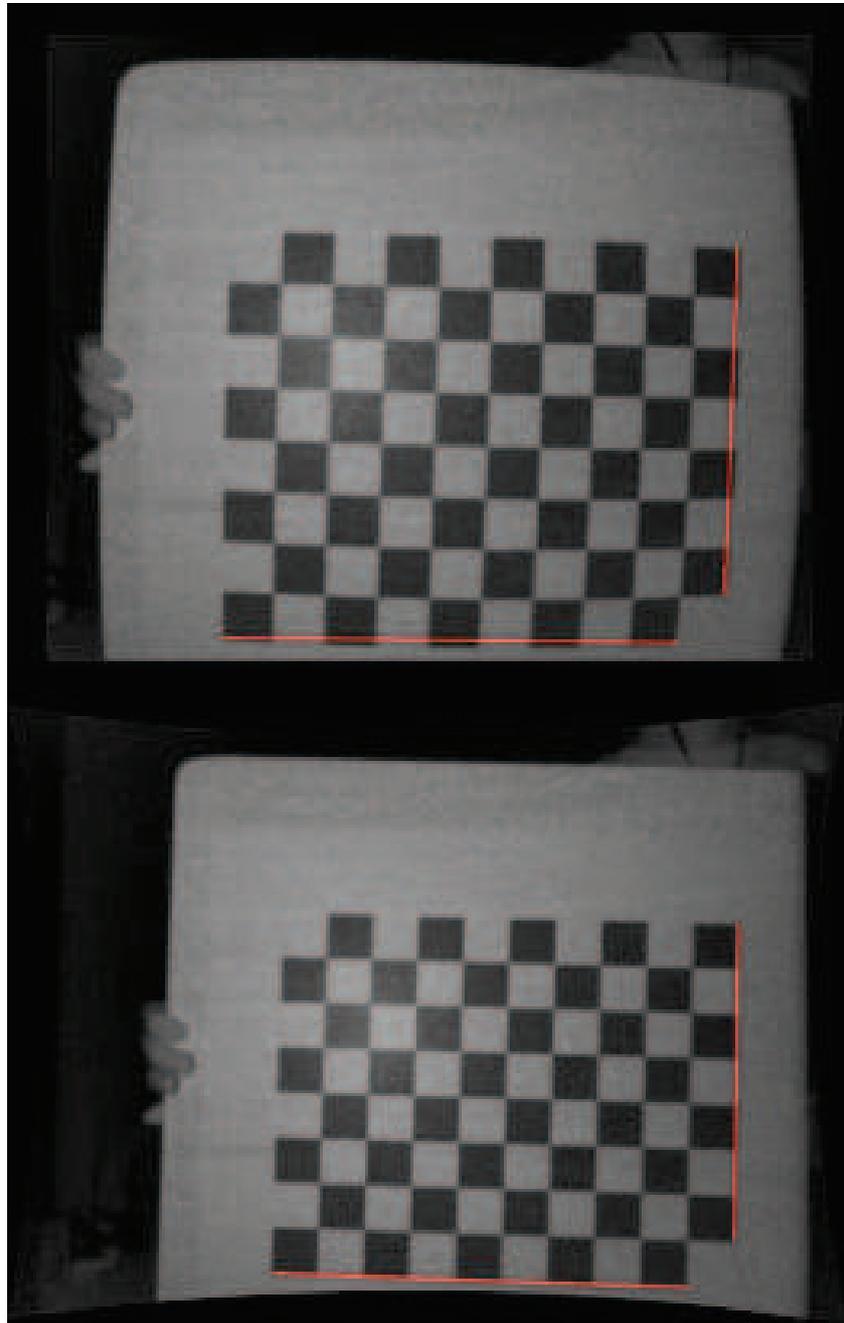


FIGURE 32 – Sur la gauche, une image du damier avant d’appliquer la correction. Sur la droite, la même image après correction de la distorsion. Les lignes rouges droites montrent l’effet de distorsion.

Pour comparer les paramètres intrinsèques que nous obtenons avec ceux du SDK d'Intel, nous utilisons l'erreur de re-projection. C'est-à-dire que nous re-projetons les coins des carrés du damier dans l'image en utilisant la matrice de la caméra du SDK et que nous comparons avec les positions de référence du damier, puis nous répétons le processus en utilisant notre matrice de caméra calculée via l'étalonnage. À la fin, nous calculons les erreurs moyennes. Voir le tableau 2 pour toutes les valeurs numériques. Nos paramètres calculés donnent une erreur de re-projection plus faible que les paramètres d'Intel.

### 6.3 Étalonnage de la profondeur

Avant d'introduire notre approche la validation de la méthode, nous renvoyons le lecteur à une évaluation approfondie du capteur Intel RealSense SR300 d'un point de vue métrologique, réalisée par Carfagni et al. [CFG<sup>+</sup>17]. Les auteurs donnent un aperçu des capacités et des limites du capteur RealSense SR300 en tant que périphérique de numérisation 3D.



FIGURE 33 – La sphère étalon utilisée dans notre processus de validation. Elle a un diamètre de 50.80 mm (2 pouces).

En conservant la même configuration matérielle que nous avons utilisée pour l'étalonnage de la profondeur, nous remplaçons le plan par une sphère comme objet étalon avec un diamètre précisément connu, voir la figure 33. Le but est de capturer la sphère à différentes positions du domaine d'étalonnage, puis d'estimer le centre de la sphère pour chaque capture via régression. Ensuite, de former une trajectoire avec les centres comme

nœuds. Pour chaque nœud de trajectoire, nous capturons deux nuages de points, l'un utilisant les valeurs d'étalonnage du SDK et l'autre utilisant nos valeurs d'étalonnage (paramètres intrinsèques inférés par le damier). Pour l'ensemble des nuages capturés en utilisant nos valeurs, nous appliquons en plus la correction de profondeur que nous avons calculé via lsetup avec la MMT.

Nous calculons deux erreurs par trajectoire, une erreur globale et une erreur locale.

### 6.3.1 L'erreur globale

Pour cet estimateur, aucune sphère de référence n'est choisie, d'où le terme global. Nous dénotons l'erreur globale  $E$ .

Nous calculons la distance de chaque centre de sphère au centre de la sphère suivante, dans l'ordre de leurs captures car aucun ordre spécifique n'est requis. Nous nous référons au premier ensemble de distances comme les distances des nuages de points et nous l'appellerons  $D_{PC}$ . De manière équivalente, nous calculons les distances entre les positions MMT successives des captures que nous appellerons distances MMT et nous dénoterons  $D_{CMM}$ . Nous définissons l'erreur globale comme suit :

$$E = \sum_{d_{PC} \in D_{PC}} \sum_{d_{CMM} \in D_{CMM}} |d_{PC} - d_{CMM}| / (num_{spheres} - 1) \quad (56)$$

Où :  $d_{CMM}$  est le correspondant de  $d_{PC}$  dans  $D_{CMM}$ .

### 6.3.2 L'erreur locale

Une erreur locale peut être calculée à chaque centre de sphère que nous avons capturé. Pour une sphère  $S$ , nous calculons l'erreur locale  $e(S)$  en prenant les distances à tous les autres centres de sphère et en les comparant aux distances respectives inférées de la MMT d'une façon similaire à l'erreur globale. L'erreur locale au centre de la sphère  $S$  est :

$$e(S) = \sum_{d_{PC} \in D_{PC}(S)} \sum_{d_{CMM} \in D_{CMM}(S)} |d_{PC} - d_{CMM}| / (num_{spheres} - 1) \quad (57)$$

Où  $D_{PC}(S)$  est l'ensemble des distances calculées à partir des nuages de points et  $D_{CMM}(S)$  est l'ensemble des distances calculées à partir des positions respectives de la MMT.  $d_{CMM}$  est le correspondant de  $d_{PC}$  dans  $D_{CMM}(S)$ .

### 6.3.3 Résultats

Nous avons capturé la sphère d'étalonnage sur vingt-sept positions différentes, comme le montre la figure 34.

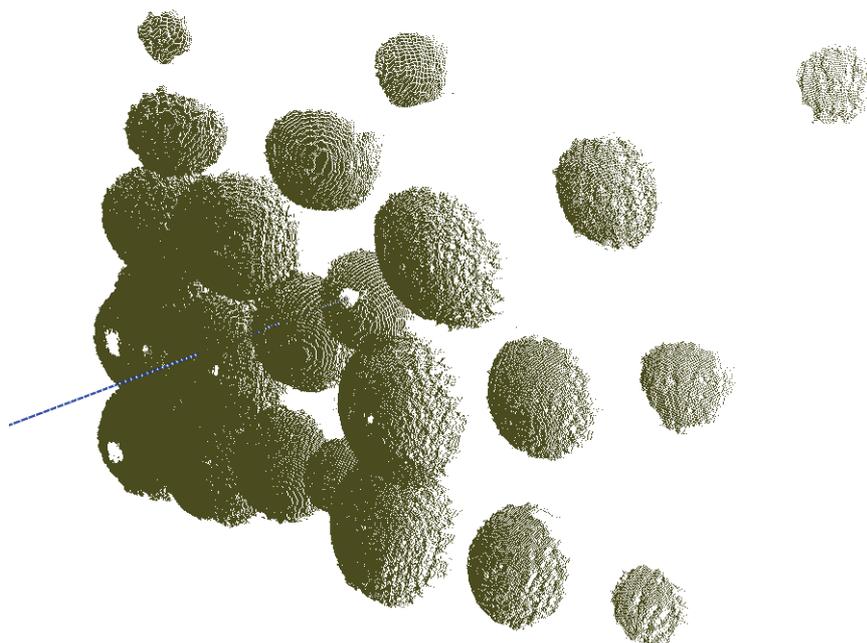


FIGURE 34 – La sphère d'étalonnage capturée sur différentes positions à l'intérieur du domaine de l'étalonnage. La ligne bleue correspond à l'axe  $Z$  du capteur.

Nous avons effectué de trois étalonnages en utilisant notre méthode dans les mêmes conditions. Les deux graphes de la figure 35 représentent les erreurs globales et locales que nous avons obtenues. Bien qu'il y ait des positions où l'étalonnage du RealSense SDK est meilleur que notre étalonnage, notre erreur globale moyenne est plus faible dans toutes les expériences, voir le tableau 2 pour l'erreur globale moyenne de chaque expérience. En ce qui concerne l'erreur locale, nous pouvons voir que notre calibration est bien meilleure que celle du SDK dans toutes les expériences.

La figure 36 montre un nuage de points avant et après l'étalonnage. Nous avons choisi un nuage de points de surface plane afin de bien voir la différence. En fait, c'est près des coins d'une surface plane couvrant tout le «cadre» que la distorsion est plus visible.

TABLE 2 – Evaluation de l'erreur globale.

	Erreur moyenne de l'étalonnage (mm)	Erreur moyenne du SDK (mm)
1er essai	0.18	0.76
2ème essai	0.27	0.76
3ème essai	0.32	0.76

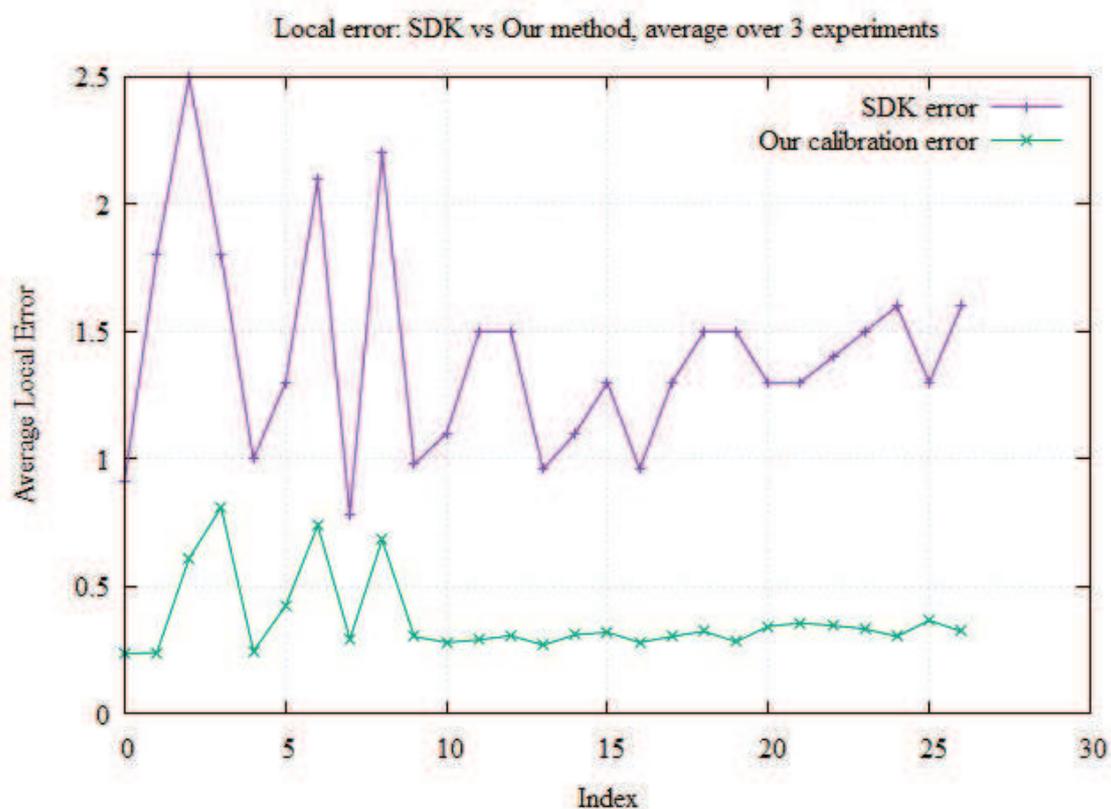
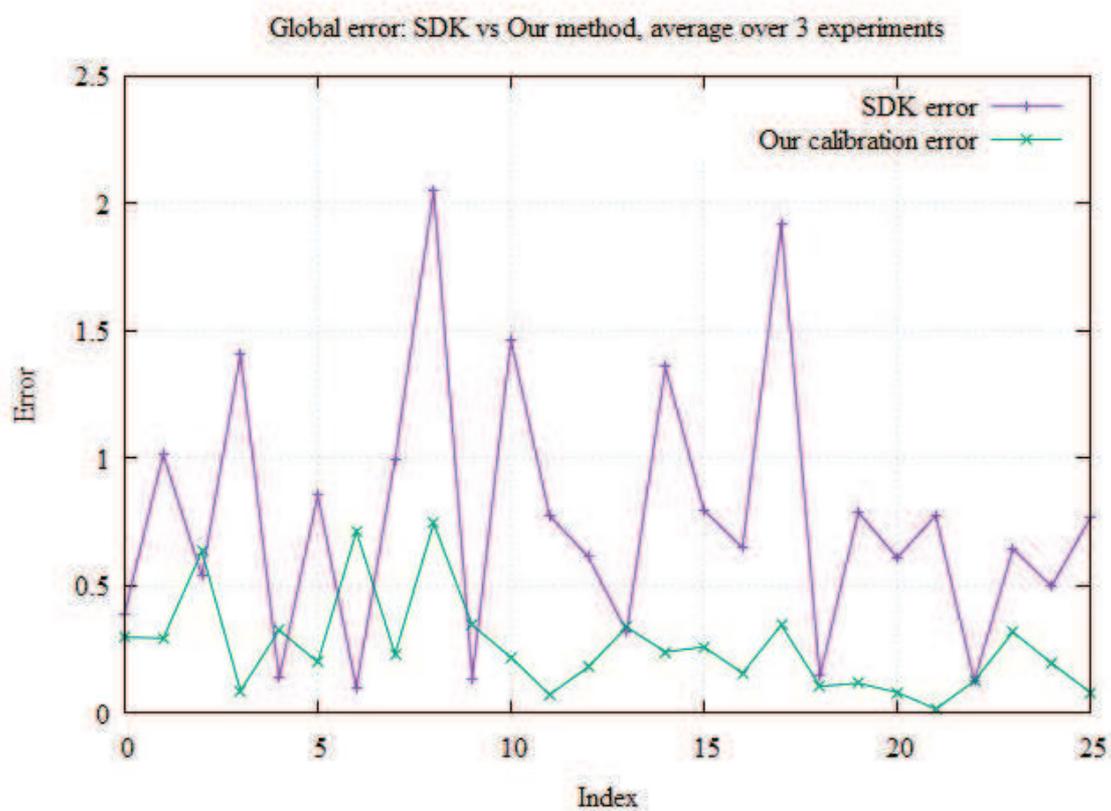


FIGURE 35 – L’erreur globale et l’erreur locale du SDK par rapport à notre méthode. Nous avons moyenné 3 expériences.

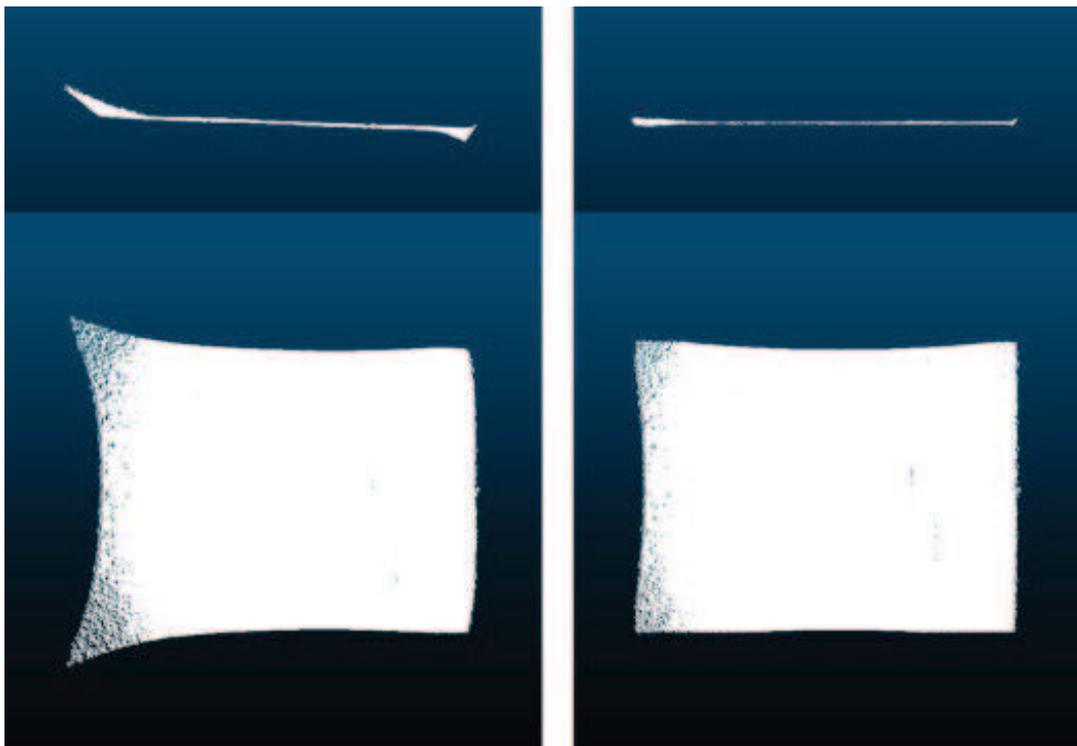


FIGURE 36 – Gauche : vue de face et de dessus d’un nuage de points (surface plane) avant correction. A droite, le même plan après correction en utilisant notre méthode.

#### 6.3.4 Quelques notes sur la précision de la méthode

La précision de notre méthode dépend essentiellement de deux facteurs :

- L’erreur de re-projection moyenne de l’étalonnage par damier (voir le tableau 2). Dans notre test, l’erreur est de 0,64 pixels.
- La précision de la distance référence calculée par traitement d’image en utilisant le principe des triangles similaires avec la distance focale calculée dans l’étalonnage 2D par damier.

Nous allons essayer d’évaluer le second facteur qui est l’erreur sur la distance référence. Elle repose fortement sur l’erreur de re-projection moyenne car les images IR corrigées et non déformées sont utilisées dans l’étape de traitement d’image pour le calcul de la distance référence.

En utilisant le principe des triangles similaires, la distance référence  $d$  est calculée comme suit :

$$d = \frac{Lf}{l} \quad (58)$$

Où,  $L$  est la demi-largeur du marqueur (en millimètres),  $l$  la demi-largeur du marqueur détecté dans l’image IR (en pixels) et  $f$  la distance focale calculée (en pixels) par l’étalonnage via damier.

Maintenant, supposons que nous fassions une erreur de  $n$  pixels dans notre détection, et que la distance calculée soit  $d$ . Ainsi, l'erreur correspondant à cette détection est approximativement :

$$E(n) \approx d - d = \frac{Lf}{l} - \frac{Lf}{l+n} \quad (59)$$

Donc,

$$E(n) \approx \frac{nLf}{l(l+n)} \quad (60)$$

La première chose que nous remarquons est que plus la valeur est grande, plus l'erreur est petite. Pour augmenter  $l$ , la caméra IR doit être réglée sur sa résolution maximale, soit 640x480 pour le RealSense SR300, et le capteur doit être très proche du plan étalon de telle sorte que le marqueur couvre la plus grande partie du cadre tout en restant entièrement visible pour réussir sa détection.

Pour avoir une idée de la précision que nous avons obtenue dans notre configuration, nous avons pu approcher le capteur du plan étalon jusqu'à atteindre 225 pixels pour  $l$ . Sachant que  $L = 79,5$  mm et  $f = 473,448$  pixels, l'erreur est :

$$E(0.64pixels) \approx 0.47mm \quad (61)$$

Ainsi, nous avons approximativement une précision d'un demi-millimètre dans notre distance référence.

## 6.4 Conclusion

Nous avons proposé une méthode d'étalonnage intrinsèque supervisée pour l'Intel RealSense SR300 qui repose sur l'utilisation d'une MMT pour calculer les distances références de manière robuste. Cette méthode nous a permis d'obtenir une précision supérieure à l'étalonnage par défaut du fabricant, comme indiqué dans la section «Résultats et validation de la méthode». En outre, nous pouvons l'appliquer à d'autres capteurs à lumière structurée, car nous n'utilisons aucun paramètre d'étalonnage spécial ou exclusif au capteur Intel RealSense SR300.

Du côté des limitations, lors du calcul des coordonnées  $X$  et  $Y$ , la méthode implique l'utilisation d'une coordonnée de profondeur qui n'est pas encore corrigée. Malgré cela, notre approche fonctionne mieux que l'étalonnage par défaut du fabricant.

Comme amélioration possible, il est possible d'effectuer un étalonnage itératif, afin d'évaluer l'erreur à chaque itération pour diminuer l'effet de la dépendance des coordonnées  $X$  et  $Y$  de la profondeur  $Z$ . D'un autre côté, il serait intéressant de pouvoir rendre notre méthode entièrement automatisée.

La partie suivante présentera une méthode pour l'acquisition de la couleur qui utilise la capture 3D pour simplifier l'estimation de BRDF.



## Troisième partie

### Acquisition de la couleur

## 1 Introduction

Le rendu réaliste d'un objet exige une étude de la réaction de la surface de cet objet vis-à-vis de la lumière. En effet, une équation dite "équation de rendu" qui est basée sur un modèle géré par des paramètres spécifiques permet d'attribuer une couleur à chaque point de l'objet.

L'équation de rendu est une équation intégrale qui exprime la luminance quittant une surface en fonction de la luminance incidente de toutes les directions, pondérée par la fonction de distribution de la réflectance bidirectionnelle BRDF. Ainsi, il s'agit essentiellement de trouver un modèle de BRDF adapté à la surface dont nous voulons générer un rendu.

Le scan 3D consiste à reconstruire la géométrie d'un objet ou d'une scène, avec l'information de la couleur dans la plupart des cas, à partir de simples images (2D) du modèle ou avec des dispositifs plus sophistiqués. Généralement, le pipeline typique d'un scan 3D est le suivant :

- L'acquisition.
- Le pré-traitement.
- Le recalage.
- Le maillage.

Notre objectif est de trouver un modèle de BRDF qui soit bien adapté au rendu d'un objet provenant d'un scanning 3D. Il s'agit essentiellement de deux étapes :

- Scan 3D des modèles de tests (nuages de points colorés).
- Recalage et filtrage des différents nuages de points et estimation des différents paramètres de la ou des BRDF choisie(s).

Nous nous restreindrons aux modèles composés d'un seul matériau

## 2 Setup matériel

Il s'agit de capturer des nuages de points colorés du modèle sous différents points de vue dans un environnement à éclairage contrôlé.

Ainsi, pour une prise de vue homogène de la couleur, nous avons opté pour l'utilisation d'une boîte à lumière qui contient 4 rampes à LED réglables individuellement, voir la figure 37.

Le capteur a été attaché de manière rigide à une lumière ponctuelle afin que la position de cette dernière soit bien connue dans toutes les captures. Cette position est mesurée par rapport à la caméra infrarouge du capteur qui constitue l'origine de repère où sont transcodées les coordonnées des points du nuage.



FIGURE 37 – Boite à lumière.

Pour l'acquisition, le modèle est placé au centre de la boîte à lumière devant le couple caméra/lumière ponctuelle voir la figure 38.

Nous tournons l'objet autour de lui-même au lieu de tourner le capteur autour. Pour chaque vue, deux nuages de points sont capturés : un avec lumière ponctuelle et un sans lumière ponctuelle.

Pour éliminer les parois de la boîte à lumière dans les nuages, nous avons utilisé le logiciel CloudCompare [url]. Voir la figure 39 pour quelques exemples. Pour nos essais, nous avons pris 56 captures au total.

### 3 Implémentation

Le programme principal a été implémenté en C++ sous l'environnement Qt sous OpenGL/GLSL. La partie du rendu est basée sur le cours [url]. Nous avons aussi fait appel à d'autres bibliothèques que nous introduisons un peu plus bas. La figure 40 montre l'interface graphique du programme. Il s'agit principalement d'une barre d'outils et d'un viewport pour la vue 3D.

La barre d'outils est constituée de :

- Un bouton « Charger » pour importer le modèle.
- Deux boutons pour régler la taille des points du nuage affiché.
- Un bouton « Optimisation » pour lancer l'estimation des paramètres de la BRDF.
- Un bouton « BRDF » pour appliquer la BRDF avec les paramètres estimés.
- Un champ « Pos. Lumière » pour modifier la position de la lumière.



FIGURE 38 – Environnement de la prise de vue : le modèle et le couple capteur/lumière ponctuelle.

## 3.1 Algorithme

L'objectif de notre algorithme est de simplifier l'estimation des paramètres de BRDF via l'estimation des normales sur les nuages de points, ainsi que la récupération directe de plusieurs observations de la couleur sur chaque point du nuage obtenu après recalage.

L'algorithme est constitué des étapes suivantes :

- Le recalage
- Le filtrage
- L'estimation des normales
- L'estimation des paramètres de la BRDF

### 3.1.1 Le recalage

**3.1.1.1 Principe** Le recalage 3D consiste à aligner deux ou plusieurs nuages de points, en général pris sous différents points de vue d'une même scène. L'objectif est d'obtenir un seul nuage de point combinant toutes les vues. En d'autres termes, il s'agit de réécrire tous les nuages dans un même repère

En fait, la caméra 3D a un repère XYZ par défaut. Par exemple, dans le cas de la caméra Intel RealSense SR300, le repère est situé au centre de la caméra infrarouge. Ainsi, les coordonnées des points retournés lors d'une capture sont exprimées dans ce repère.

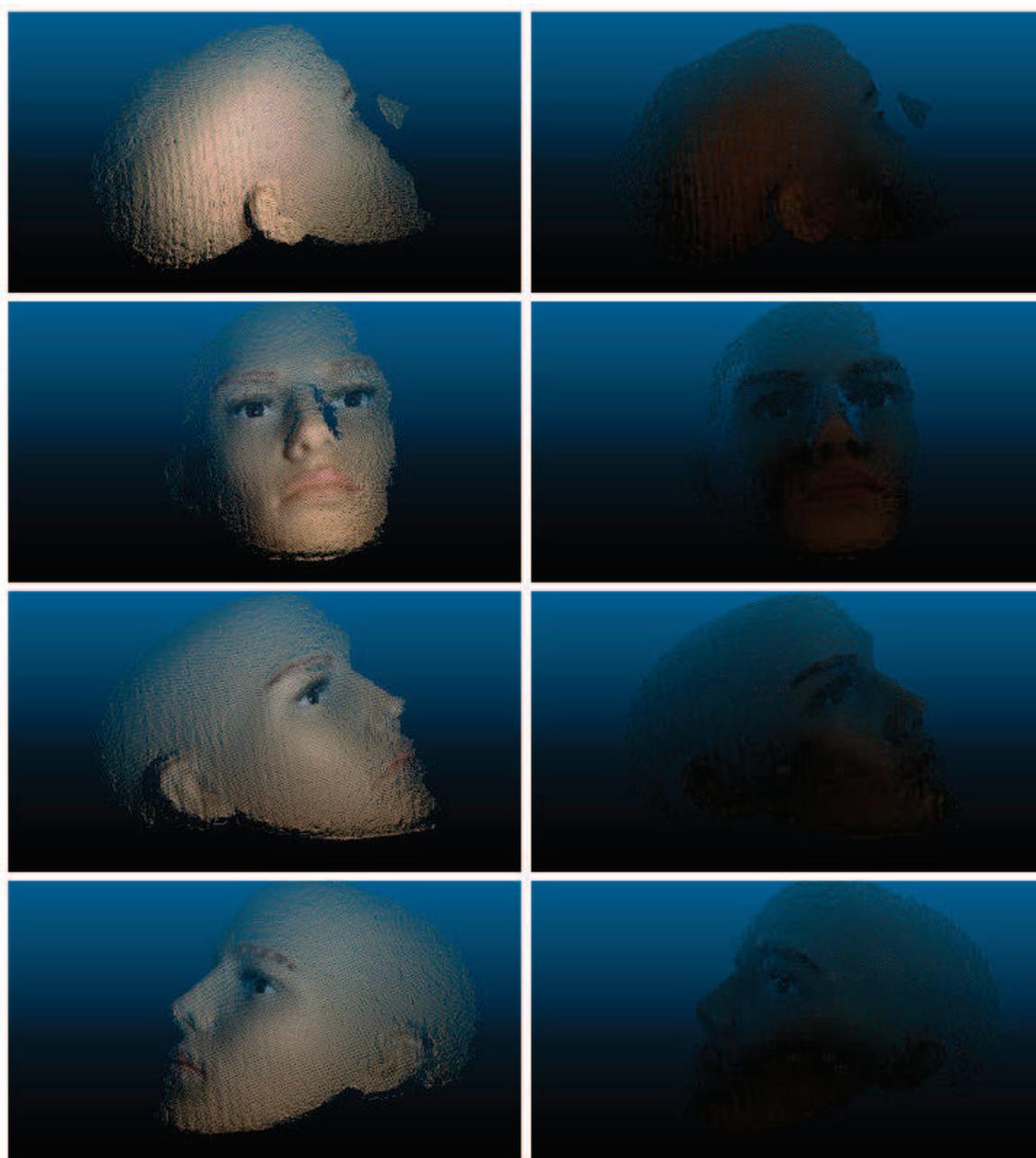


FIGURE 39 – Exemple de captures sous différents points de vue avec et sans lumières

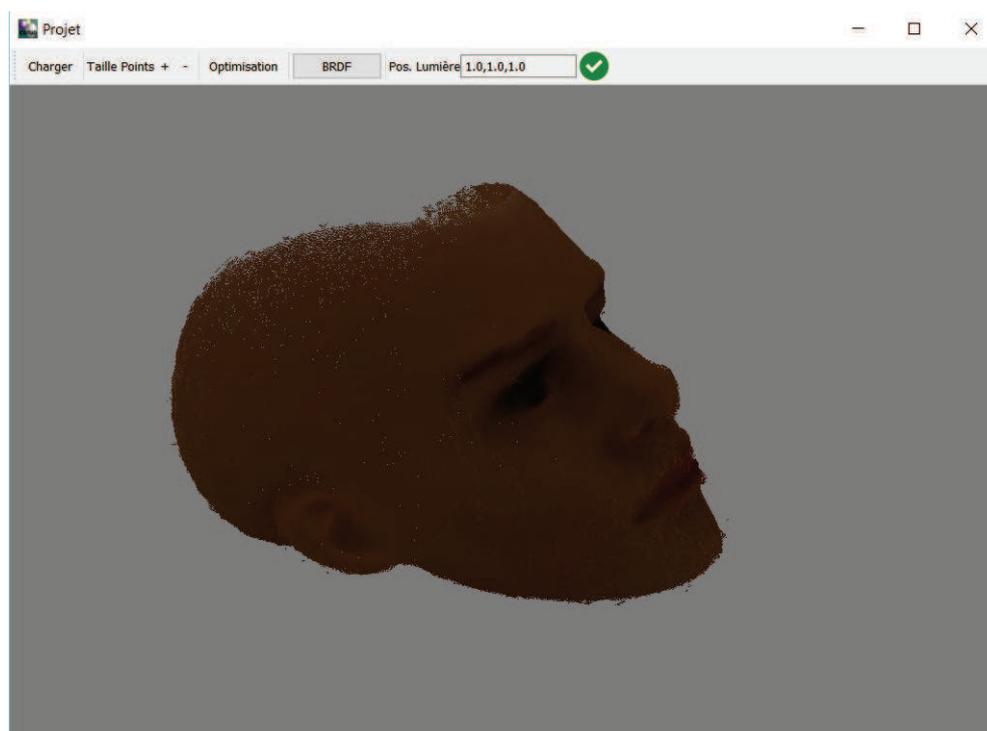


FIGURE 40 – Interface graphique du programme principal

Ainsi, pour deux nuages de points nous cherchons une transformation  $T$  de type  $T(x) = Rx + t$  où  $R$  est une rotation et  $t$  une translation, permettant de transcoder les points de l'un des deux nuages dans le système des coordonnées de l'autre. Voir la figure 42.

Pour le recalage, nous avons utilisé la bibliothèque Libpointmatcher [urlh], écrite en C++, qui implémente l'algorithme ICP (Iterative Closest Point).

**3.1.1.2 L'algorithme ICP** L'algorithme ICP (Iterative Closest Point) a été introduit par Chen et Medioni en 1991 [CM92] et Besl et McKay en 1992 [BM92] pour le recalage de certains types de représentations géométriques de données, dont les nuages de points font partie. L'algorithme ICP consiste en 4 étapes [HIT<sup>+</sup>15] :

- La sélection ou échantillonnage dans les deux nuages de points à recalcer
- La recherche de correspondants entre les deux nuages (ré-échantillonnés)
- Le filtrage de faux correspondants
- L'estimation de la transformation (alignement) qui consiste à assigner une métrique d'erreur pour ensuite minimiser l'erreur globale

**3.1.1.2.1 La sélection** La sélection nous permet à extraire les parties « utiles » au recalage dans un nuage de points provenant directement du capteur. En effet, une information redondante, des points aberrants, ou un très grand nombre de points peuvent alourdir inutilement le temps de calcul ainsi que les ressources utilisées lors du processus

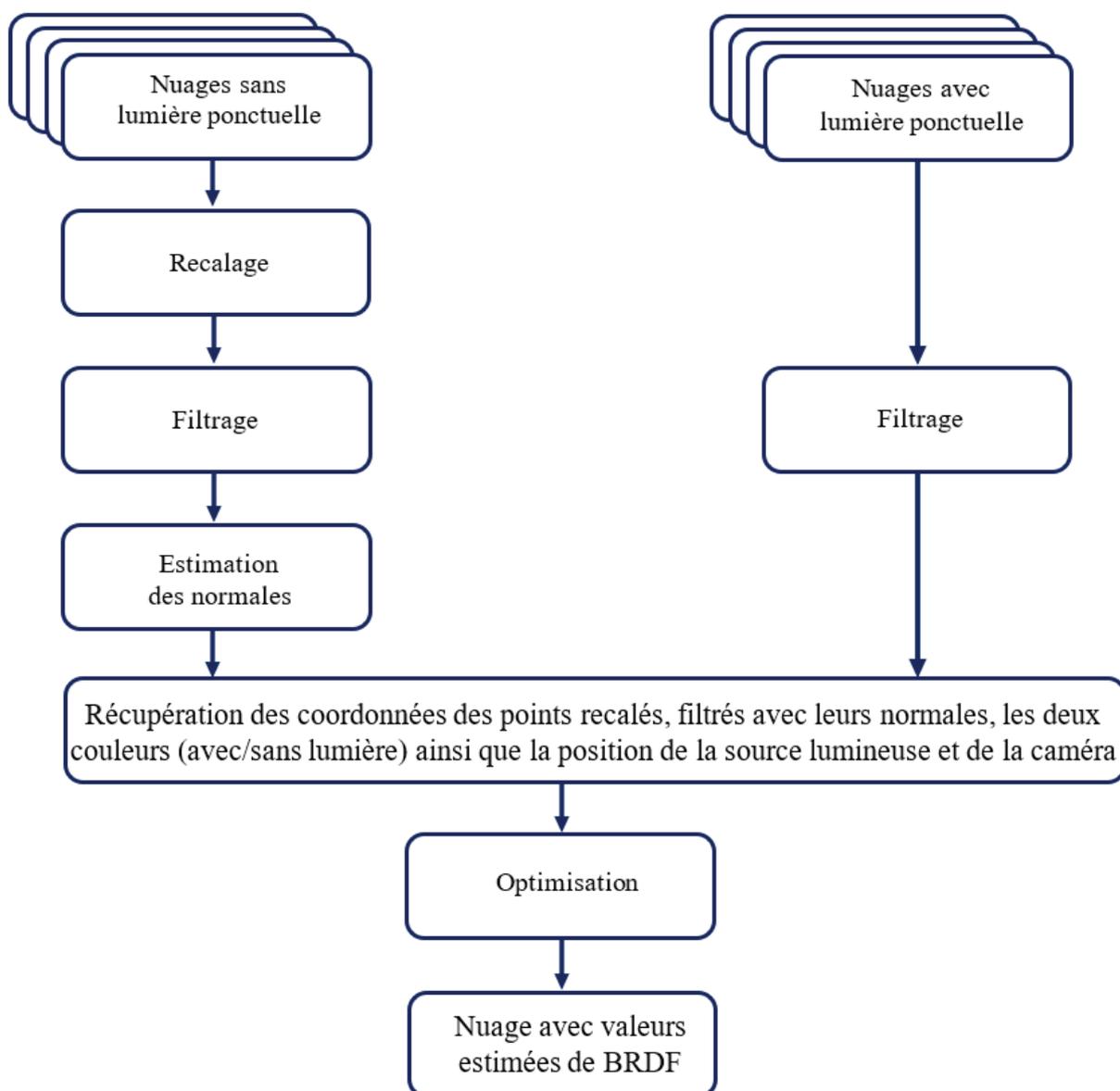


FIGURE 41 – Enchaînement de notre algorithme

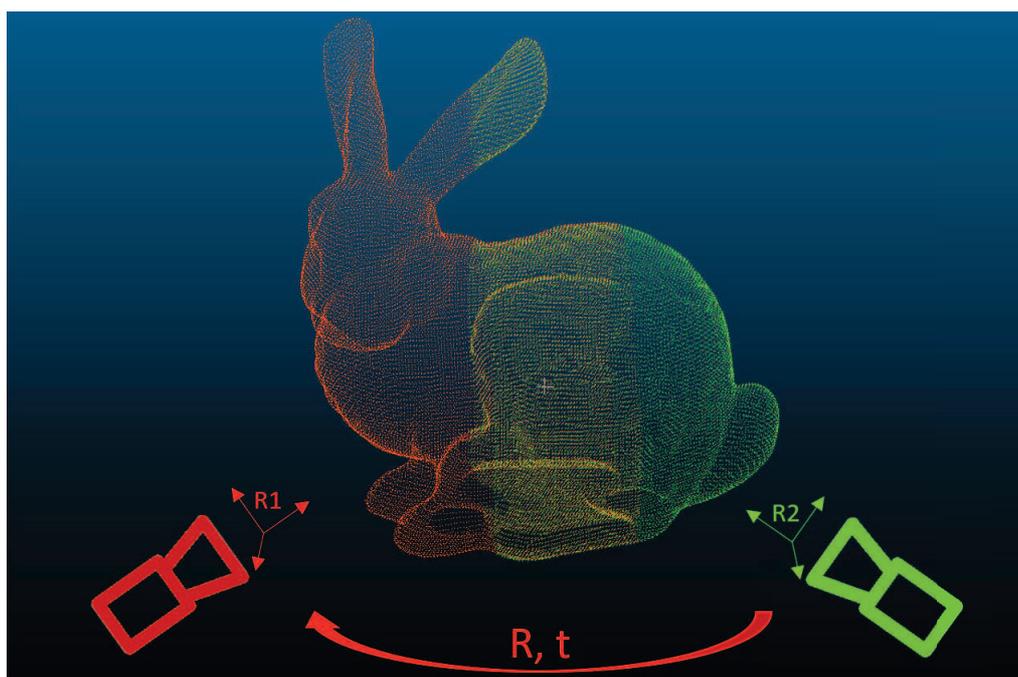


FIGURE 42 – Exemple de Recalage 3D entre deux captures. Nous considérons le repère  $R1$  de la 1ère prise de vue comme référence et il faut trouver la rotation  $R$  et la translation  $t$  (matrice de transformation) pour passer du repère  $R2$  vers le repère  $R1$

de recalage [HIT<sup>+</sup>15].

Certains algorithmes de filtrage de points aberrants ou/et de ré-échantillonnage peuvent être utilisés lors de la sélection. Voici quelques exemples :

- Pass-through filter : il reste l'une des manières les plus simples et les plus directes de filtrage rapide d'un nuage de points. Il s'agit d'éliminer tout point du nuage dont l'une des coordonnées est située en dehors d'un intervalle préalablement fixé. Voir la figure 43.

- Elimination de points aberrants par critère de voisinage (sphérique) : L'élimination par critère de voisinage sphérique ou radius outlier removal en anglais supprime les points qui n'ont pas un certain nombre minimal de voisins sur un rayon donné. Ceci permet d'enlever les parties qui ne sont pas dense du nuage de points. Ces parties peuvent compromettre l'estimation des caractéristiques locales comme les normales. Voir la figure 44.

- Sous-échantillonnage aléatoire : Ce type de sous-échantillonnage permet de réduire le nombre de points du nuage selon une valeur de probabilité qu'on fixe. En effet, cette valeur représente le pourcentage des points que nous voulons enlever/garder. La densité des points dans le nuage est aussi réduite de façon uniforme.

- Sous-échantillonnage par grille de voxels : Cette technique permet de réduire le nombre d'échantillons dans le nuage de points. Il s'agit de diviser l'espace par une grille de voxels de tailles égales, et de remplacer chaque ensemble de points appartenant à un

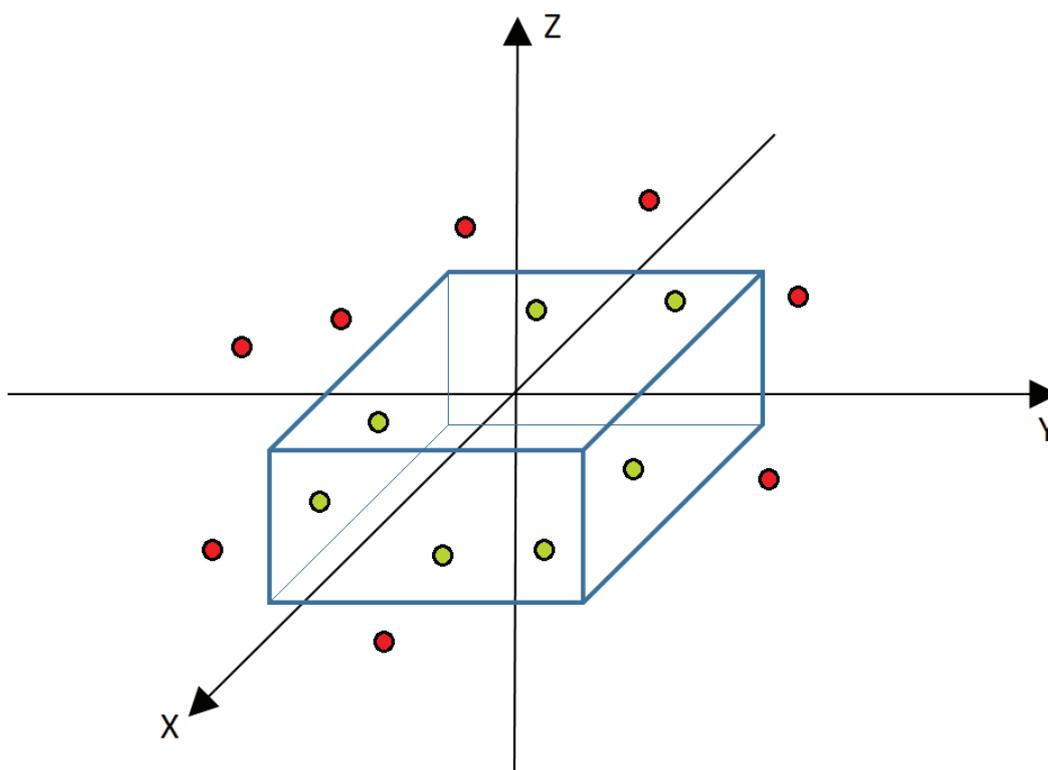


FIGURE 43 – Exemple d'un pass-through filter avec la définition d'intervalles bornés pour les trois axes  $X$ ,  $Y$  et  $Z$ . Ainsi, on définit une boîte dans l'espace de telle sorte que chaque point se trouvant à l'extérieur de la boîte est éliminé.

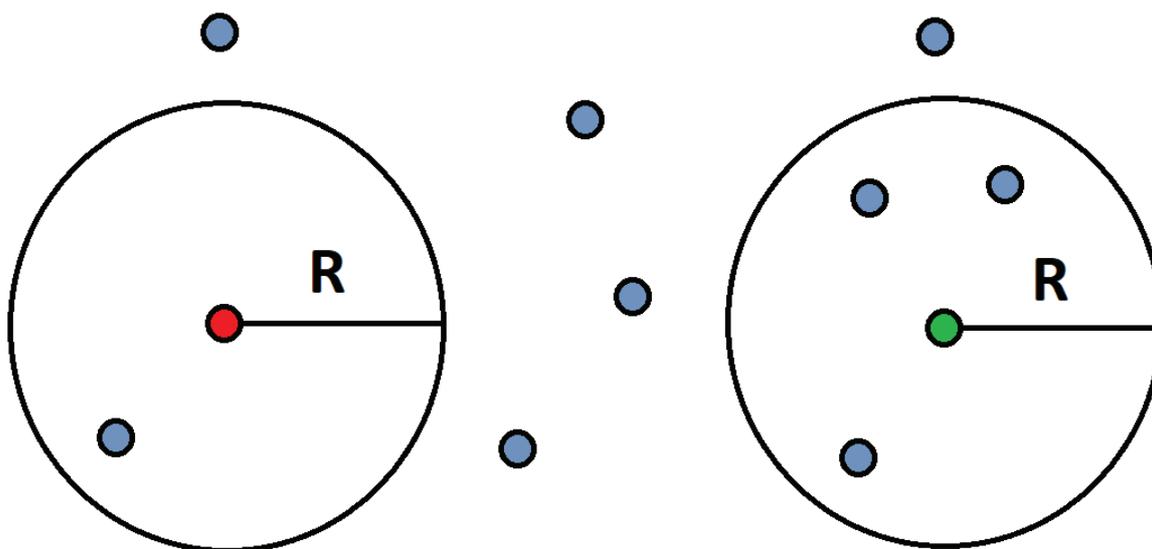


FIGURE 44 – Elimination par critère de voisinage sphérique. Dans cet exemple, on élimine les points qui ont moins de deux voisins sur un rayon  $R$ . Le point à gauche (rouge) est éliminé alors que celui de droite (vert) reste.

voxel par son barycentre. Voir la figure 45.

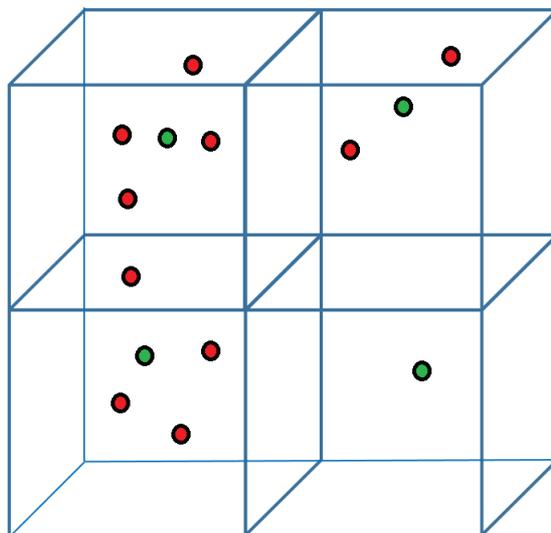


FIGURE 45 – Sous-échantillonnage par grille de voxels. Chaque sous-nuage de points appartenant à un voxel est remplacé par son barycentre.

**3.1.1.2.2 La recherche de correspondants** Cette étape consiste à chercher pour chaque point d'un des nuages à recaler un correspondant dans l'autre nuage afin d'établir des couples de points entre les deux nuages. Un couple de points représente le même point de l'espace vu de deux positions différentes.

La recherche de correspondants peut se faire de plusieurs manières. La plus simple consiste à chercher pour chaque point d'un nuage le point le plus proche dans l'autre nuage. Cette recherche peut devenir assez conséquente en termes de ressources, ainsi, l'utilisation d'une structure de données comme les kd-tree devient nécessaire. Une deuxième méthode consiste à projeter suivant le point de vue de l'autre nuage pour trouver un correspondant [urlk].

**3.1.1.2.3 Le filtrage de faux correspondants** Il s'agit d'éliminer les faux correspondants ou au moins limiter leur influence en leur affectant des poids permettant de contrôler leurs contributions dans les calculs.

Plusieurs choix de poids peuvent être utilisés comme la distance entre les deux points du correspondant par rapport à la distance moyenne de tous les correspondants, l'angle entre les deux normales constituant la paire... [HIT<sup>+</sup>15]

**3.1.1.2.4 L'estimation de la transformation (alignement)** Il faut commencer par choisir une métrique afin de quantifier l'erreur pour pouvoir ensuite la minimiser.

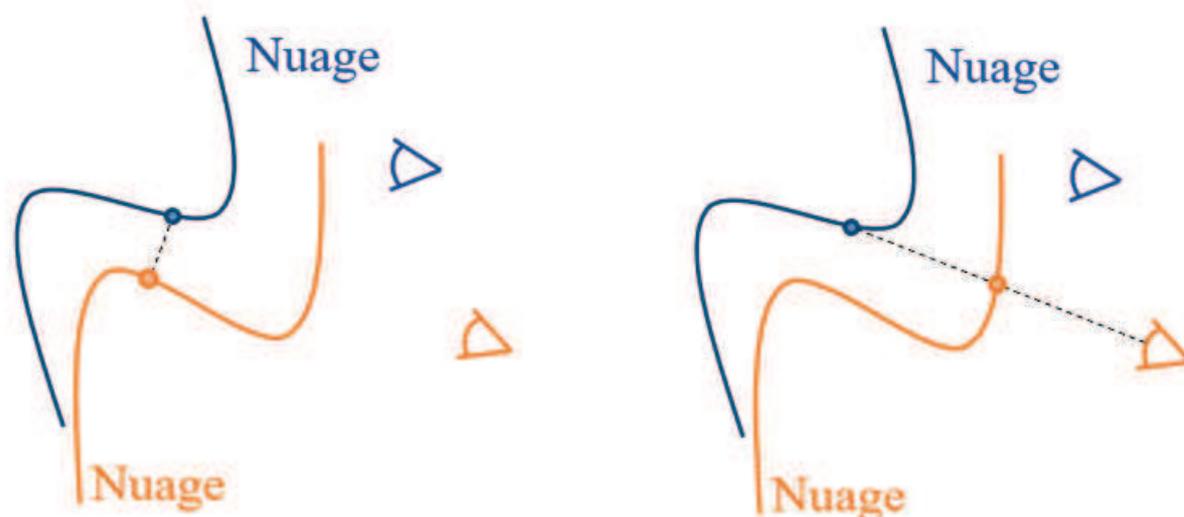


FIGURE 46 – Recherche de correspondants : A gauche, recherche du point le plus proche et à droite la projection selon le point de vue de l'autre nuage.

### Erreur point-à-point

L'erreur dite « point-à-point » s'exprime comme suit :

$$E_{point-a-point}(T) = \sum_{k=1}^N \|Tp_k - q_k\|^2 \quad (62)$$

Avec  $N$  le nombre de correspondants  $(p_k, q_k)$  et  $T$  la transformation que nous cherchons.

Il s'agit de la somme des distances entre les points de chaque correspondant. Ainsi, en minimisant cette erreur, on réduit l'écart entre les deux nuages.

### Erreur point-à-plan

L'erreur dite « point-à-plan » s'exprime comme suit :

$$E_{point-plan}(T) = \sum_{k=1}^N \omega_k ((Tp_k - q_k) \cdot n_{q_k})^2 \quad (63)$$

Avec  $N$  le nombre de correspondants  $(p_k, q_k)$ ,  $T$  la transformation qu'on cherche et  $n_{q_k}$  la normale dans  $q_k$  par rapport à son voisinage.

Après avoir défini la métrique, vient l'étape d'optimisation afin de minimiser l'erreur. Au final, nous obtenons une transformation de type rotation/translation.

**3.1.1.3 Algorithme** ICP est un algorithme de recalage par pair (de nuage de points). En outre, l'algorithme exige que les deux nuages soient assez proches et aient une zone commune, pour la recherche de correspondants (étape 2 de l'algorithme), afin d'être

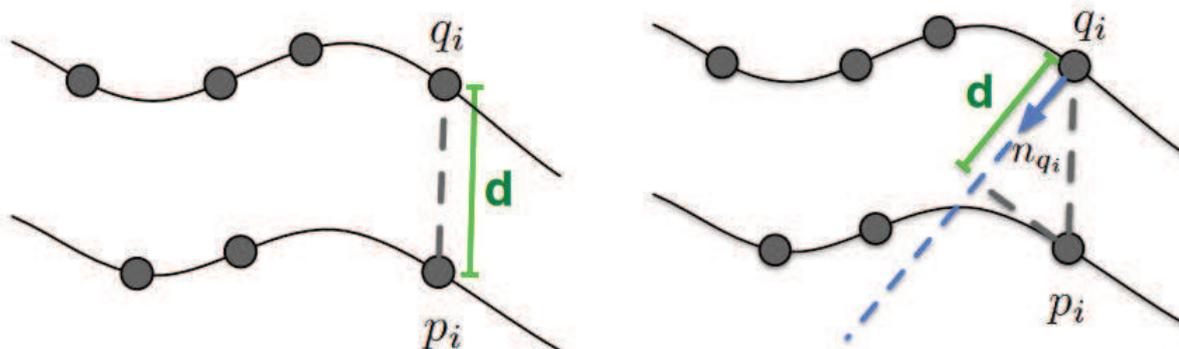


FIGURE 47 – Métriques d’erreur. À gauche la métrique point-à-point et à droite la métrique point-à-plan [HIT<sup>+</sup>15].

recalés. Donc, pour recalcr l’ensemble des captures du modèle, on a procédé comme suit :

- Recalage de chaque nuage avec celui qui le précède : Pour chaque nuage  $N_i$  nous obtenons une transformation  $M_i$ , représentant une rotation et une translation. Le premier nuage est considéré comme recalé sur lui-même, ainsi,  $M_0$  est l’identité.

- Recalage de chaque nuage avec le premier : La transformation  $T_k$  du nuage  $N_k$  est :

$$T_k = \prod_{i=0}^k M_i \quad (64)$$

Pour  $n$  nuages de points, nous aurons un ensemble de couples  $(N_k, T_k), k = 1..n$ . La figure 48 montre un exemple de recalage à plusieurs nuages de points.

### 3.1.2 Filtrage

Concernant le filtrage, nous avons utilisé l’algorithme d’élimination de points aberrants par critère de voisinage sphérique, que nous avons déjà introduit précédemment. La recherche de voisins a été faite avec la bibliothèque nanoflann [urln] écrite en C++ et qui implémente la structure Kd-tree.

### 3.1.3 Estimation des normales

Dans le but d’estimer les normales, nous avons travaillé avec la bibliothèque nanoflann. Pour chaque point, nous recherchons les voisins dans un rayon donné, puis nous régressons par un plan. Pour la régression, nous avons utilisé la fonction `getBestFitPlane` récupérée de la bibliothèque Bullet Physics SDK [urlb]. La normale du plan trouvé est considérée comme normale au point traité.

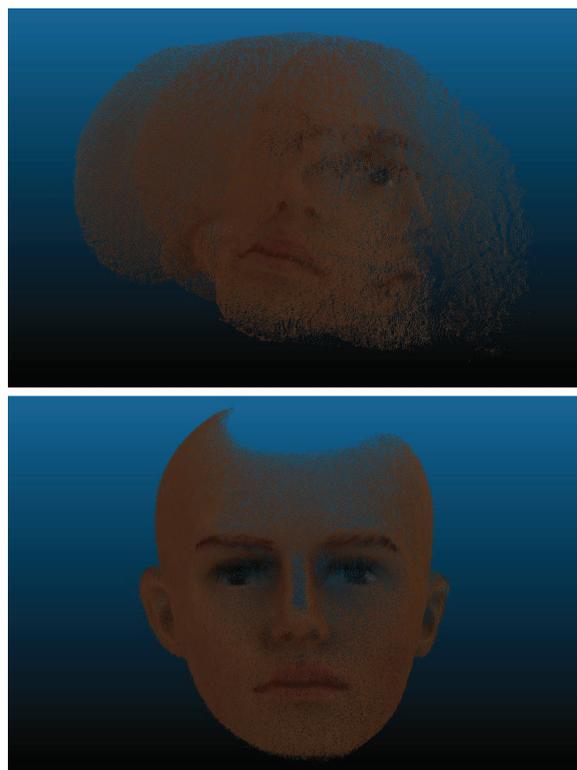


FIGURE 48 – Exemple de nuages de points avant recalage en haut (4 nuages), et exemple après recalage en bas (28 nuages).



FIGURE 49 – Exemple de filtrage avec un minimum de 15 voisins sur un rayon de 1 mm sur le nuage final (recalé). A droite le nuage avant et à gauche le nuage après filtrage.

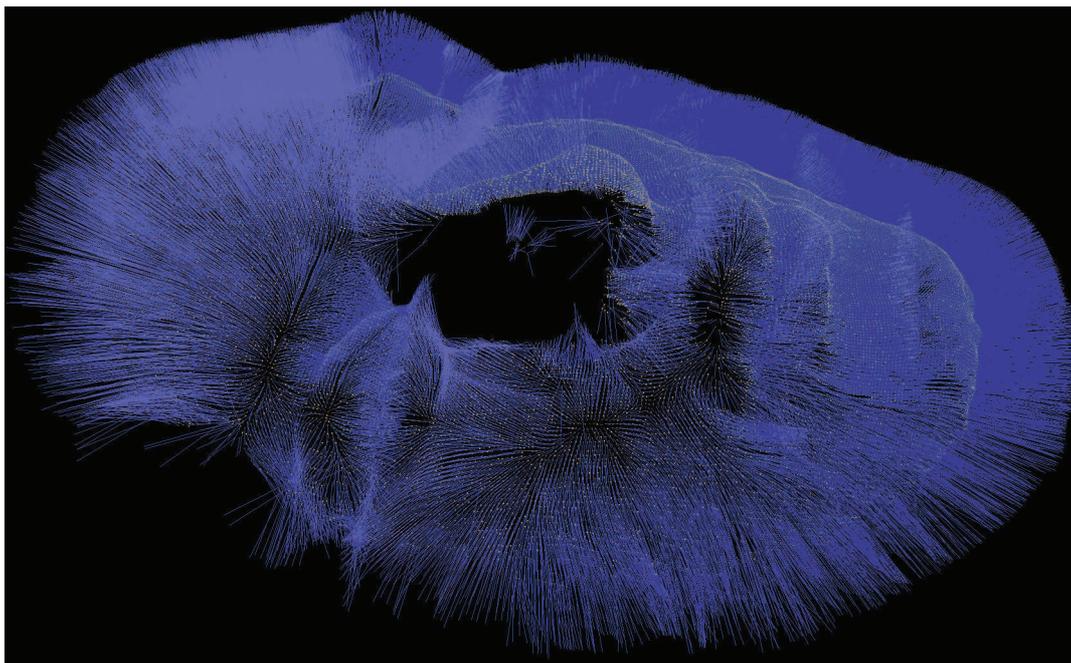


FIGURE 50 – Exemple de nuage de points avec les normales estimées (affichés dans le logiciel [urlr])

### 3.1.4 Estimation des différents paramètres de la ou des BRDF choisie(s)

Pour estimer les valeurs des différents paramètres de la BRDF, nous utilisons l’algorithme de Levenberg-Marquardt [Gav11] qui est un algorithme itératif pour la résolution de problèmes d’optimisation de fonctions dépendants de plusieurs paramètres. Cet algorithme est implémenté dans la bibliothèque lmdif [urlm] qui a été écrite à l’origine en fortran puis portée en langage C.

Nous partons d’un ensemble de nuages de points du même objet, pris sous différents points de vue avec la position de la lumière et du capteur 3D connues pour chaque nuage. Du coup, après l’étape du recalage, en termes de couleur, chaque point du nuage final se retrouve avec un certain nombre de voisins qui représentent en théorie le même point pris sous différentes conditions (lumière et angle de vision), c’est-à-dire différentes mesures (de couleur) du même point.

Ainsi, nous devons résoudre le problème non-linéaire des moindres carrés suivant :

$$\min_p \sum_{i=1}^m \omega_k (C_i - \hat{C}_i)^2 \quad (65)$$

Où :

- $m$  est le nombre d’observations (mesures) du même point
- $p$  représente les paramètres que nous cherchons
- $C_i$  la couleur calculée en utilisant la BRDF

-  $\hat{C}_i$  la couleur observée (sur les nuages capturés avec lumière).

Le problème (65) n'est pas résolu sur tous les points du nuage. Pour qu'un point soit éligible, nous lui exigeons un nombre minimal de mesures afin d'assurer une bonne estimation des paramètres de la BRDF que nous cherchons.

Afin de calculer la couleur  $C_i$  pour chaque point éligible, nous commençons par récupérer sa couleur sur les captures prises sans lumière ponctuelle (figure 39) à laquelle nous rajoutons la BRDF calculée avec les paramètres de l'itération courante de l'algorithme Levenberg-Marquardt.

Etant donné que l'algorithme de Levenberg-Marquardt est itératif, pour vérifier la stabilité des résultats par rapport au choix des valeurs initiales, nous varions les paramètres de la BRDF choisie dans leurs intervalles respectifs.

## 3.2 Algorithme

### 3.2.1 Résultats Cook-Torrance

D'après les équations (43), (44) et (45), la couleur  $C_i$  de la formule à minimiser (65) est la suivante :

$$C_i = k f_{lambert} + (1 - k) \frac{F.G}{4(N.L)(N.V)} \frac{1}{\pi m^2 (N.H)^4} e^{\left(\frac{(N.H)^2 - 1}{m^2 (N.H)^2}\right)} \quad (66)$$

Avec  $k = k_d + k_s = 1$

Les paramètres que nous cherchons à estimer sont  $m$  et  $k$  qui varient entre 0 et 1 [urlg].

TABLE 3 – Modèle de Cook-Torrance : résultats obtenus pour différentes valeurs initiales  $m_0$  et  $k_0$ .  $m$  et  $k$  sont les moyennes de chaque paramètre sur les points optimisés.

	$k_0 = 0.2$	$k_0 = 0.5$	$k_0 = 0.8$
$m_0 = 0.2$	$m = 0.351324$ $k = 0.710756$	$m = 0.377163$ $k = 0.70719$	$m = 0.400712$ $k = 0.7104$
$m_0 = 0.5$	$m = 0.527524$ $k = 0.699864$	$m = 0.520772$ $k = 0.682537$	$m = 0.550313$ $k = 0.696551$
$m_0 = 0.8$	$m = 0.677881$ $k = 0.660926$	$m = 0.675166$ $k = 0.635755$	$m = 0.611807$ $k = 0.702619$

La figure 51 montre un exemple de résultat après l'étape de l'optimisation (dans le cas  $m_0 = 0.8$  et  $k_0 = 0.5$ ) comparée à l'image réelle prise avec lumière. Dans le rendu, nous avons utilisé la même position de lumière que celle utilisé lors de la prise de vue.

Dans les résultats du tableau 3, nous remarquons que le paramètre  $k$  est généralement stable, alors que le paramètre  $m$  reste très proche de sa valeur initiale. Ceci peut-être expliqué par le fait que l'équation de rendu (66) est linéaire en  $k$  alors qu'elle ne l'est pas en  $m$ .

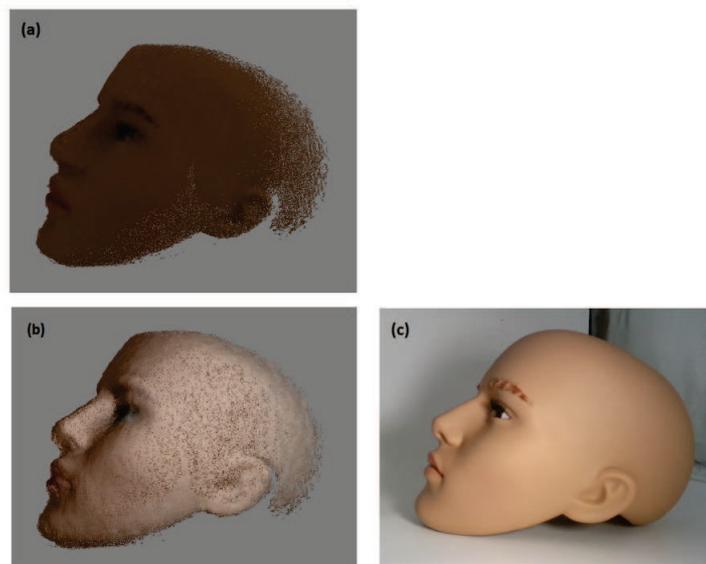


FIGURE 51 – (a) Rendu direct avec la couleur des captures sans lumières, (b) Rendu avec les paramètres estimés de Cook-Torrance (c) Image Réelle

Pour le rendu final, nous remarquons que les paramètres estimés de Cook-Torrance donnent un résultat proche du modèle réel. Voir la figure 51.

### 3.2.2 Résultats Ward

D'après l'équation (42), la couleur  $C_i$  de la formule à minimiser (65) est la suivante :

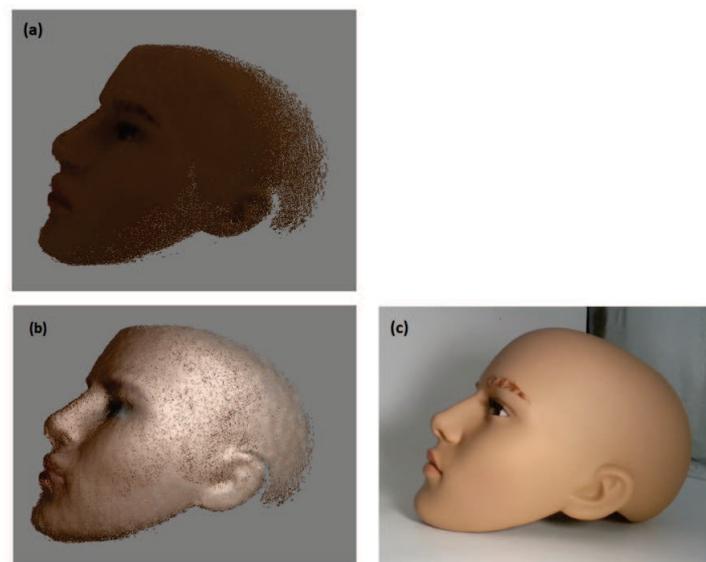


FIGURE 52 – (a) Rendu direct avec la couleur des captures sans lumières, (b) Rendu avec les paramètres estimés de Ward (c) Image Réelle

TABLE 4 – Modèle de Ward : résultats obtenus pour différentes valeurs initiales  $m_x0$ ,  $m_y0$  et  $k_0$ .  $m_x$ ,  $m_y$  et  $k$  sont les moyennes de chaque paramètre sur les points optimisés.

	$k_0 = 0.2$	$k_0 = 0.5$	$k_0 = 0.8$
$m_{x0} = 0.2$ $m_{y0} = 0.2$	$m_x = 0.225022$ $m_y = 0.232243$ $k = 0.711916$	$m_x = 0.230812$ $m_y = 0.234276$ $k = 0.702601$	$m_x = 0.246772$ $m_y = 0.244804$ $k = 0.742542$
$m_{x0} = 0.2$ $m_{y0} = 0.5$	$m_x = 0.192448$ $m_y = 0.385002$ $k = 0.705075$	$m_x = 0.213903$ $m_y = 0.374532$ $k = 0.699772$	$m_x = 0.21125$ $m_y = 0.368734$ $k = 0.745305$
$m_{x0} = 0.2$ $m_{y0} = 0.8$	$m_x = 0.192489$ $m_y = 0.426951$ $k = 0.693838$	$m_x = 0.203357$ $m_y = 0.430211$ $k = 0.69733$	$m_x = 0.20718$ $m_y = 0.484381$ $k = 0.746382$
$m_{x0} = 0.5$ $m_{y0} = 0.2$	$m_x = 0.360022$ $m_y = 0.206821$ $k = 0.718317$	$m_x = 0.392813$ $m_y = 0.249302$ $k = 0.706118$	$m_x = 0.37844$ $m_y = 0.259152$ $k = 0.745246$
$m_{x0} = 0.5$ $m_{y0} = 0.5$	$m_x = 0.334011$ $m_y = 0.367349$ $k = 0.677479$	$m_x = 0.342143$ $m_y = 0.378841$ $k = 0.676462$	$m_x = 0.357101$ $m_y = 0.373141$ $k = 0.743662$
$m_{x0} = 0.5$ $m_{y0} = 0.8$	$m_x = 0.353641$ $m_y = 0.441332$ $k = 0.659362$	$m_x = 0.382357$ $m_y = 0.458747$ $k = 0.650354$	$m_x = 0.372402$ $m_y = 0.482936$ $k = 0.729476$
$m_{x0} = 0.8$ $m_{y0} = 0.2$	$m_x = 0.396873$ $m_y = 0.229344$ $k = 0.70836$	$m_x = 0.431451$ $m_y = 0.256416$ $k = 0.701053$	$m_x = 0.467853$ $m_y = 0.253348$ $k = 0.746091$
$m_{x0} = 0.8$ $m_{y0} = 0.5$	$m_x = 0.419732$ $m_y = 0.414003$ $k = 0.648767$	$m_x = 0.406312$ $m_y = 0.449963$ $k = 0.645788$	$m_x = 0.439739$ $m_y = 0.410582$ $k = 0.731629$
$m_{x0} = 0.8$ $m_{y0} = 0.8$	$m_x = 0.451576$ $m_y = 0.52404$ $k = 0.57626$	$m_x = 0.474343$ $m_y = 0.536032$ $k = 0.59881$	$m_x = 0.476842$ $m_y = 0.524226$ $k = 0.693677$

$$C_i = \frac{k}{\pi} + (1 - k) \frac{1}{\sqrt{\cos\theta_i \cos\theta_r}} \frac{1}{4\pi m_x m_y} e^{-2 \frac{(\frac{H_x}{m_x})^2 + (\frac{H_y}{m_y})^2}{1 + (H \cdot N)}} \quad (67)$$

Les paramètres que nous cherchons à estimer sont  $m_x$ ,  $m(y)$  et  $k$  qui varient entre 0 et 1 [urlg].

La figure 52 montre un exemple de résultat après l'étape de l'optimisation (dans le cas  $m_x0 = 0.8$ ,  $m_y0 = 0.8$  et  $k_0 = 0.5$ ) comparée à l'image réelle prise avec lumière. Dans le rendu, nous avons utilisé la même position de lumière que celle utilisée lors de la prise de vue.

Dans le tableau 4, on remarque que les résultats pour le paramètre  $k$  sont plus stables que les résultats pour les deux paramètres  $m_x$  et  $m_y$ . Ceci peut-être expliqué par le fait

que l'équation de rendu (67) est linéaire en  $k$  alors qu'elle ne l'est pas en  $m_x$  et  $m_y$ .

Pour le rendu final, nous remarquons que les paramètres estimés de Ward donnent un résultat éloigné du modèle réel. nous remarquons plus de spécularité sur la figure 52 (b) comparée à la figure 52 (c).

### 3.2.3 Résultats Kelemen et Szirmay-Kalos

Nous avons ajouté ce modèle qui est plus récent que les autres BRDF testées afin de comparer. Il date de 2001. C'est un modèle physiquement valide.

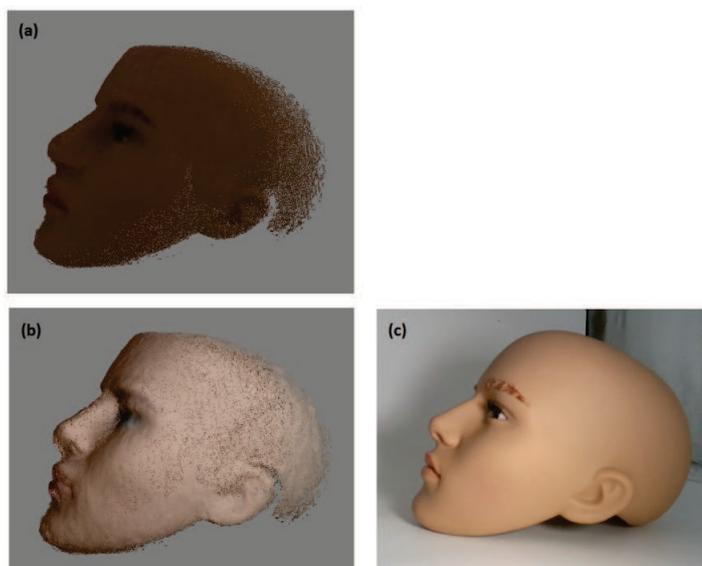


FIGURE 53 – (a) Rendu direct avec la couleur des captures sans lumières, (b) Rendu avec les paramètres estimés de Kelemen et Szirmay-Kalos (c) Image Réelle

La partie spéculaire du modèle est la suivante [KSK01] :

$$f_{specular} = \frac{F}{h \cdot h} \frac{1}{\pi m^2 (N \cdot H)^4} e^{\left(\frac{(N \cdot H)^2 - 1}{m^2 (N \cdot H)^2}\right)} \quad (68)$$

Où :  $h$  est la direction spéculaire non normalisée (somme de la direction de la lumière ainsi que la direction de la caméra).

Du coup, la couleur  $C_i$  de la formule à minimiser (65) est la suivante :

$$C(i) = k f_{lambert} + (1 - k) f_{specular} \quad (69)$$

Les paramètres que nous cherchons à estimer sont  $m$  et  $k$  qui varient entre 0 et 1 [urlg].

La figure 53 montre un exemple de résultat après l'étape de l'optimisation (dans le cas  $m_0 = 0.8$  et  $k_0 = 0.5$ ) comparée à l'image réelle prise avec lumière. Dans le rendu, nous avons utilisé la même position de lumière que celle utilisée lors de la prise de vue.

TABLE 5 – Modèle de Kelemen et Szirmay-Kalos : résultats obtenus pour différentes valeurs initiales  $m_0$  et  $k_0$ .  $m$  et  $k$  sont les moyennes de chaque paramètre sur les points optimisés.

	$k_0 = 0.2$	$k_0 = 0.5$	$k_0 = 0.8$
$m_0 = 0.2$	$m = 0.718212$ $k = 0.392467$	$m = 0.615382$ $k = 0.660325$	$m = 0.474865$ $k = 0.761284$
$m_0 = 0.5$	$m = 0.678547$ $k = 0.570894$	$m = 0.685718$ $k = 0.643699$	$m = 0.559142$ $k = 0.760838$
$m_0 = 0.8$	$m = 0.721948$ $k = 0.56905$	$m = 0.738114$ $k = 0.645283$	$m = 0.681847$ $k = 0.760926$

Dans les résultats du tableau 5, nous remarquons que les paramètres  $m$  et  $k$  restent généralement stables, contrairement à l'estimation avec le modèle de Cook-Torrance et Ward.

Pour le rendu final, nous remarquons que les paramètres estimés de Kelemen et Szirmay-Kalos donnent un résultat proche du modèle réel, voir la figure 53.



# Conclusion générale

Cette thèse a été pour moi une occasion de faire de la recherche au sein de l'entreprise, permettant d'évoluer dans un cadre hybride entre théorie et pratique. Ainsi, outre les résultats présentés dans ce manuscrit, j'ai pu travailler sur deux projets industriels dont l'un portant sur un scan 3d complet d'objets pour le calcul d'une boîte englobante optimale et l'autre portant spécifiquement sur le recalage dans un cadre bien défini.

Nous rappelons que ce travail est axé sur deux thématiques principales : l'acquisition de la géométrie ainsi que l'acquisition de la couleur. Le matériel visé étant les capteurs ou caméras 3D grand public. Cette catégorie étant principalement représentée par les capteurs utilisant la technologie lumière structurée, nous avons basé nos travaux dessus. Ainsi, nous avons pu travailler sur deux générations successives de capteurs 3D Intel de la famille RealSense, qui représente le mieux selon nous cette catégorie de capteurs.

Dans le thème acquisition de la géométrie, après un premier projet industriel sur le scan 3D le calcul d'une boîte englobante optimale, ainsi qu'un deuxième projet se focalisant sur le recalage, nous avons constaté un grand besoin pour travailler sur l'étalonnage de capteurs qui représente selon nous un grand handicap pour les capteurs 3D grand public surtout quand on vise des application industrielles. Ainsi, Nous avons proposé une méthode d'étalonnage intrinsèque supervisée pour l'Intel RealSense SR300 qui repose sur l'utilisation d'une MMT pour calculer les distances références de manière robuste. Cette méthode nous a permis d'obtenir une précision supérieure à l'étalonnage par défaut du fabricant, comme indiqué dans la section «Résultats et validation de la méthode». En outre, nous pouvons l'appliquer à d'autres capteurs à lumière structurée, car nous n'utilisons aucun paramètre d'étalonnage spécial ou exclusif au capteur Intel RealSense SR300.

Quant à l'acquisition de la couleur, nous avons développé une méthode qui tient compte de la géométrie de l'objet afin de faciliter l'estimation d'une BRDF compatible. Le but étant de pouvoir intégrer cette technique dans un scanner pour visage, nous avons travaillé avec une tête de mannequin dans un environnement contrôlé pour valider nos résultats.

Ce travail pourra encore faire l'objet de recherche et développement. Nous pouvons citer notamment l'automatisation de la méthode d'étalonnage ainsi que l'acquisition de couleur dans un environnement moins contrôlé et avec des objets composés de plusieurs matériaux/couleurs.

# Bibliographie

- [AFS<sup>+</sup>11] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10) :105–112, 2011.
- [Bla04] François Blais. Review of 20 years of range sensor development. *Journal of electronic imaging*, 13(1) :231–244, 2004.
- [Bli77] James F Blinn. Models of light reflection for computer synthesized pictures. In *ACM SIGGRAPH computer graphics*, volume 11, pages 192–198. ACM, 1977.
- [BM92] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV : Control Paradigms and Data Structures*, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.
- [BMNK13] Kai Berger, Stephan Meister, Rahul Nair, and Daniel Kondermann. A state of the art report on kinect sensor setups in computer vision. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 257–272. Springer, 2013.
- [BTVG06] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf : Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.
- [CFG<sup>+</sup>17] Monica Carfagni, Rocco Furferi, Lapo Governi, Michaela Servi, Francesca Uccheddu, and Yary Volpe. On the performance of the intel sr300 depth camera : metrological and critical characterization. *IEEE Sensors Journal*, 17(14) :4508–4519, 2017.
- [CM92] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3) :145–155, 1992.
- [CMNK13] Dan A Calian, Kenny Mitchell, Derek Nowrouzezahrai, and Jan Kautz. The shading probe : Fast appearance acquisition for mobile ar. In *SIGGRAPH Asia 2013 Technical Briefs*, page 20. ACM, 2013.
- [Com87] International Electrotechnical Commission. Radiometric, photometric and colorimetric measurements : physical detectors. 1987. Section 845-05.
- [CT82] Robert L Cook and Kenneth E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics (TOG)*, 1(1) :7–24, 1982.
- [Cur00] Brian Curless. Overview of active vision techniques. In *Proc. SIGGRAPH*, volume 99, 2000.

- [DRS10] Julie Dorsey, Holly Rushmeier, and François Sillion. *Digital modeling of material appearance*. Elsevier, 2010.
- [FB87] Martin A Fischler and Robert C Bolles. Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. In *Readings in computer vision*, pages 726–740. Elsevier, 1987.
- [FH<sup>+</sup>15] Yasutaka Furukawa, Carlos Hernández, et al. Multi-view stereo : A tutorial. *Foundations and Trends® in Computer Graphics and Vision*, 9(1-2) :1–148, 2015.
- [GAMNGL<sup>+</sup>09] Diego González-Aguilera, Angel Muñoz-Nieto, Javier Gómez-Lahoz, Jesus Herrero-Pascual, and Gabriel Gutierrez-Alonso. 3d digital surveying and modelling of cave geometry : Application to paleolithic rock art. *Sensors*, 9(2) :1108–1127, 2009.
- [Gav11] Henri Gavin. The levenberg-marquardt method for nonlinear least squares curve-fitting problems. *Department of Civil and Environmental Engineering, Duke University*, pages 1–15, 2011.
- [Hei00] Janne Heikkila. Geometric camera calibration using circular control points. *IEEE Transactions on pattern analysis and machine intelligence*, 22(10) :1066–1077, 2000.
- [Hel25] H.V. Helmholtz. *Treatise on Physiological Optics*, volume 1. Dover :New York, 1925.
- [Hil] Stephen Hill. SIGGRAPH 2012 Course : Practical Physically Based Shading in Film and Game Production - Self Shadow. [Online ; accessed 28. Mar. 2018]. URL : <https://blog.selfshadow.com/publications/s2012-shading-course>.
- [HIT<sup>+</sup>15] Dirk Holz, Alexandru E Ichim, Federico Tombari, Radu B Rusu, and Sven Behnke. Registration with the point cloud library : A modular framework for aligning in 3-d. *IEEE Robotics & Automation Magazine*, 22(4) :110–124, 2015.
- [HKH<sup>+</sup>12] Daniel Herrera, Juho Kannala, Janne Heikkilä, et al. Joint depth and color camera calibration with distortion correction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(10) :2058–2064, 2012.
- [HP10] Klaus Häming and Gabriele Peters. The structure-from-motion reconstruction pipeline—a survey with focus on short image sequences. *Kybernetika*, 46(5) :926–937, 2010.

- [HZ05] Richard Hartley and Andrew Zisserman. Multiple view geometry in computer vision. *Robotica*, 23(2) :271–271, 2005.
- [INK<sup>+</sup>] Shahram Izadi, Richard A Newcombe, David Kim, Otmar Hilliges, David Molyneaux, Steve Hodges, Pushmeet Kohli, Jamie Shotton, Andrew J Davison, and Andrew Fitzgibbon. Kinectfusion : real-time dynamic 3d surface reconstruction and interaction, acm siggraph 2011 talks, august 07-11, 2011, vancouver. *British Columbia, Canada*[doi:10.1145/2037826.2037857].
- [IY01] Gavriel J Iddan and Giora Yahav. Three-dimensional imaging in the studio and elsewhere. In *Three-Dimensional Image Capture and Applications IV*, volume 4298, pages 48–56. International Society for Optics and Photonics, 2001.
- [JLG14] Bingwen Jin, Hao Lei, and Weidong Geng. Accurate intrinsic calibration of depth camera with cuboids. In *European Conference on Computer Vision*, pages 788–803. Springer, 2014.
- [Kar13] Branko Karan. Accuracy improvements of consumer-grade 3d sensors for robotic applications. In *Intelligent Systems and Informatics (SISY), 2013 IEEE 11th International Symposium on*, pages 141–146. IEEE, 2013.
- [Kar15] Branko Karan. Calibration of kinect-type rgb-d sensors for robotic applications. *FME Transactions*, 43(1) :47–54, 2015.
- [KBKL09] Andreas Kolb, Erhardt Barth, Reinhard Koch, and Rasmus Larsen. Time-of-flight sensors in computer graphics. In *Eurographics (STARs)*, pages 119–134, 2009.
- [KE12] Kouros Khoshelham and Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2) :1437–1454, 2012.
- [KSK01] Csaba Kelemen and Laszlo Szirmay-Kalos. A microfacet based coupled specular-matte brdf model with importance sampling. In *Eurographics Short Presentations*, volume 2, page 4, 2001.
- [KTTW12] Martin Knecht, Georg Tanzmeister, Christoph Traxler, and Michael Wimmer. Interactive brdf estimation for mixed-reality applications. 2012.
- [LJB<sup>+</sup>13] Joon-Young Lee, Jiyoung Jung, Yunsu Bok, Jaesik Park, Dong-Geol Choi, Yudeog Han, and In So Kweon. Robust computer vision techniques for high-quality 3d modeling. In *Pattern Recognition (ACPR), 2013 2nd IAPR Asian Conference on*, pages 6–10. IEEE, 2013.

- [LNL<sup>+</sup>13] Damien Lefloch, Rahul Nair, Frank Lenzen, Henrik Schäfer, Lee Streeter, Michael J Cree, Reinhard Koch, and Andreas Kolb. Technical foundation and calibration methods for time-of-flight cameras. In *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, pages 3–24. Springer, 2013.
- [Low04] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2) :91–110, 2004.
- [LT09] Douglas Lanman and Gabriel Taubin. Build your own 3d scanner : 3d photography for beginners. In *ACM SIGGRAPH 2009 Courses*, page 8. ACM, 2009.
- [MVG<sup>+</sup>10] Theo Moons, Luc Van Gool, Maarten Vergauwen, et al. 3d reconstruction from multiple images part 1 : Principles. *Foundations and trends® in Computer Graphics and Vision*, 4(4) :287–404, 2010.
- [MWA<sup>+</sup>13] Przemyslaw Musialski, Peter Wonka, Daniel G Aliaga, Michael Wimmer, L v Gool, and Werner Purgathofer. A survey of urban reconstruction. In *Computer graphics forum*, volume 32, pages 146–177. Wiley Online Library, 2013.
- [Nic65] Fred E Nicodemus. Directional reflectance and emissivity of an opaque surface. *Applied optics*, 4(7) :767–775, 1965.
- [Noe99] Nicolas Noe. *Etude de fonctions de distribution de la réflectance bidirectionnelle*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne ; Université Jean Monnet-Saint-Etienne, 1999.
- [OFT13] Hannes Ovrén, Per-Erik Forssén, and David Törnqvist. Why would i want a gyroscope on my rgb-d sensor? In *Robot Vision (WORV), 2013 IEEE Workshop on*, pages 68–75. IEEE, 2013.
- [Pho75] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6) :311–317, 1975.
- [SBMM15] Aaron N Staranowicz, Garrett R Brown, Fabio Morbidi, and Gian-Luca Mariottini. Practical and accurate calibration of rgb-d cameras using spheres. *Computer Vision and Image Understanding*, 137 :102–114, 2015.
- [SCD<sup>+</sup>06] Steven M Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 519–528. IEEE, 2006.

- [Sch94] Christophe Schlick. An inexpensive brdf model for physically-based rendering. In *Computer graphics forum*, volume 13, pages 233–246. Wiley Online Library, 1994.
- [Sem16] Oleksandr Semeniuta. Analysis of camera calibration with respect to measurement accuracy. 2016.
- [SFPL10] Joaquim Salvi, Sergio Fernandez, Tomislav Pribanic, and Xavier Llado. A state of the art in structured light patterns for surface profilometry. *Pattern recognition*, 43(8) :2666–2680, 2010.
- [SJP13] Jan Smisek, Michal Jancosek, and Tomas Pajdla. 3d with kinect. In *Consumer depth cameras for computer vision*, pages 3–25. Springer, 2013.
- [SM99] Peter F Sturm and Stephen J Maybank. On plane-based camera calibration : A general algorithm, singularities, applications. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, volume 1, pages 432–437. IEEE, 1999.
- [Sze10] Richard Szeliski. *Computer vision : algorithms and applications*. Springer Science & Business Media, 2010.
- [urla] Archived - Intel® RealSense™ Data Ranges | Intel® Software. [Online; accessed 17. May 2018]. URL : <https://software.intel.com/en-us/articles/intel-realsense-data-ranges>.
- [urlb] Bullet Physics SDK. [Online; accessed 28. Mar. 2018]. URL : <https://github.com/bulletphysics>.
- [urlc] Camera Calibration Toolbox for Matlab. [Online; accessed 17. May 2018]. URL : [http://www.vision.caltech.edu/bouguetj/calib\\_doc/index.html](http://www.vision.caltech.edu/bouguetj/calib_doc/index.html).
- [urld] Camera calibration With OpenCV. OpenCV 2.4.13.6 documentation. [Online; accessed 2. Jul. 2018]. URL : [https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera\\_calibration/camera\\_calibration.html](https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html).
- [urle] CloudCompare - Open Source project. [Online; accessed 28. Mar. 2018]. URL : <http://www.danielgm.net/cc>.
- [urlf] Cnam - Laboratoire commun de métrologie - Accueil rayonnements optiques. [Online; accessed 28. Mar. 2018]. URL : <http://inm.cnam.fr/accueil-rayonnements-optiques-974310.kjsp>.
- [urlg] Coding Labs - Physically Based Rendering - Cook-Torrance. [Online; accessed 28. Mar. 2018]. URL : [http://www.codinglabs.net/article\\_physically\\_based\\_rendering\\_cook\\_torrance.aspx](http://www.codinglabs.net/article_physically_based_rendering_cook_torrance.aspx).

- [urlh] ethz-asl/libpointmatcher. [Online; accessed 28. Mar. 2018]. URL : <https://github.com/ethz-asl/libpointmatcher>.
- [urli] HS Group - Group. [Online; accessed 28. Mar. 2018]. URL : [https://www.human-solutions.com/group/front\\_content.php?changelang=2&lang=2](https://www.human-solutions.com/group/front_content.php?changelang=2&lang=2).
- [urlj] Intel RealSense Camera SR300 - Product Datasheet. [Online; accessed 28. Mar. 2018]. URL : <https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/realsense-sr300-datasheet1-0.pdf>.
- [urlk] Introduction to Mobile Robotics - Iterative Closest Point Algorithm. [Online; accessed 28. Mar. 2018]. URL : <https://cs.gmu.edu/~kosecka/cs685/cs685-icp.pdf>.
- [urll] Jean-Michel DISCHLER - Support de cours : Equation du rendu : illumination globale. [Online; accessed 28. Mar. 2018]. URL : [https://dpt-info.u-strasbg.fr/~dischler/enseignement/Rendu\\_et\\_visu/EquationRendu.pdf](https://dpt-info.u-strasbg.fr/~dischler/enseignement/Rendu_et_visu/EquationRendu.pdf).
- [urlm] jeremybarnes/cephes. [Online; accessed 28. Mar. 2018]. URL : <https://github.com/jeremybarnes/cephes/blob/master/linalg/lmdif.c>.
- [urln] jlblancoc/nanoflann. [Online; accessed 28. Mar. 2018]. URL : <https://github.com/jlblancoc/nanoflann>.
- [urlo] 'Kinect for Xbox 360' is Official Name of Microsoft's Controller-Free Game Device | Stories. [Online; accessed 22. Mar. 2018]. URL : <https://news.microsoft.com/2010/06/13/kinect-for-xbox-360-is-official-name-of-microsofts-controller-free-game-device>.
- [urlp] Kinect Technology in Games | SIGGRAPH 2014. [Online; accessed 22. Mar. 2018]. URL : <http://s2014.siggraph.org/attendees/courses/events/kinect-technology-games.html>.
- [urlq] La 3D avec le module OpenGL de Qt. [Online; accessed 28. Mar. 2018]. URL : <https://cpp.developpez.com/redaction/data/pages/users/gbdivers/qtopengl>.
- [urlr] MeshLab. [Online; accessed 28. Mar. 2018]. URL : <http://www.meshlab.net>.
- [urls] Official Structure Sensor Store - Give Your iPad 3D Vision. [Online; accessed 2. Jul. 2018]. URL : <https://structure.io>.

- [urlt] RVP vision probe for REVO-2. [Online; accessed 28. Mar. 2018]. URL : <http://www.renishaw.com/media/pdf/en/e79719fe3e3746ae847cff491720efe4.pdf>.
- [urlu] Sony Depthsensing Solutions > Home. [Online; accessed 22. Mar. 2018]. URL : <https://www.sony-depthsensing.com/Default.aspx>.
- [urlv] Structure Sensor - 3D scanning, augmented reality, and more for mobile devices. [Online; accessed 22. Mar. 2018]. URL : <https://structure.io>.
- [urlw] Tango. [Online; accessed 22. Mar. 2018]. URL : <https://get.google.com/tango>.
- [urlx] Virtual Fashion. [Online; accessed 22. Mar. 2018]. URL : <http://www.virtualfashion.jp/en/index.html>.
- [War92] Gregory J Ward. Measuring and modeling anisotropic reflection. In *ACM SIGGRAPH Computer Graphics*, volume 26, pages 265–272. ACM, 1992.
- [WCH<sup>+</sup>92] Juyang Weng, Paul Cohen, Marc Herniou, et al. Camera calibration with distortion models and accuracy evaluation. *IEEE Transactions on pattern analysis and machine intelligence*, 14(10) :965–980, 1992.
- [WMLT07] Bruce Walter, Stephen R Marschner, Hongsong Li, and Kenneth E Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pages 195–206. Eurographics Association, 2007.
- [Zha00] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11) :1330–1334, 2000.
- [Zha04] Zhengyou Zhang. Camera calibration with one-dimensional objects. *IEEE transactions on pattern analysis and machine intelligence*, 26(7) :892–899, 2004.
- [ZTCS99] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading : a survey. *IEEE transactions on pattern analysis and machine intelligence*, 21(8) :690–706, 1999.

***Publications et communications relatives à ce travail***

R. Chakib, N. Mérillou, P.-J. Vincent, S. Mérillou, "Calibrating Low-cost Structured-light 3D Sensors", WSCG 2018.

## Acquisition et rendu 3D réaliste à partir de périphériques «grand public»

**Résumé :** L'imagerie numérique, de la synthèse d'images à la vision par ordinateur est entrain de connaître une forte évolution, due entre autres facteurs à la démocratisation et au succès commercial des caméras 3D. Dans le même contexte, l'impression 3D grand public, qui est en train de vivre un essor fulgurant, contribue à la forte demande sur ce type de caméra pour les besoins de la numérisation 3D.

L'objectif de cette thèse est d'acquérir et de maîtriser un savoir-faire dans le domaine de la capture/acquisition de modèles 3D en particulier sur l'aspect rendu réaliste. La réalisation d'un scanner 3D à partir d'une caméra RGB-D fait partie de l'objectif.

Lors de la phase d'acquisition, en particulier pour un dispositif portable, on est confronté à deux problèmes principaux, le problème lié au référentiel de chaque capture et le rendu final de l'objet reconstruit.

**Mots clés :** Numérisation 3D , nuage de point 3D, caméra à lumière structurée, étalonnage de caméra, caméra RGB-D, modèle sténopé, étalonnage intrinsèque, recalage de nuage de points, BRDF.

## Capture and Realistic 3D rendering from consumer grade devices

**Abstract :** Digital imaging, from the synthesis of images to computer vision is experiencing a strong evolution, due among other factors to the democratization and commercial success of 3D cameras. In the same context, the consumer 3D printing, which is experiencing a rapid rise, contributes to the strong demand for this type of camera for the needs of 3D scanning.

The objective of this thesis is to acquire and master a know-how in the field of the capture / acquisition of 3D models in particular on the rendered aspect. The realization of a 3D scanner from a RGB-D camera is part of the goal.

During the acquisition phase, especially for a portable device, there are two main problems, the problem related to the repository of each capture and the final rendering of the reconstructed object.

**Keywords :** 3D scanning, 3D point cloud, structured-light camera, camera calibration, RGB-D camera, pinhole model, intrinsic calibration, point cloud registration, BRDF.