Université de Montréal

Reconstruction active par projection de lumière non structurée

par

Nicolas Martin

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

Thèse présentée à la Faculté des arts et des sciences
en vue de l'obtention du grade de
Philosophiae Doctor (Ph.D.)
en informatique

Avril, 2014

À mon père, qui, sans le savoir, a fait de moi ce que
je suis depuis mon plus jeune âge.
À ma mère, qui m'a toujours soutenu, peu importe
combien c'était difficile.
Cette thèse est le résultat de vos encouragements
et de votre confiance en moi.

RÉSUMÉ

Cette thèse porte sur la reconstruction active de modèles 3D à l'aide d'une caméra et d'un projecteur. Les méthodes de reconstruction standards utilisent des motifs de lumière codée qui ont leurs forces et leurs faiblesses. Nous introduisons de nouveaux motifs basés sur la lumière non structurée afin de pallier aux manques des méthodes existantes. Les travaux présentés s'articulent autour de trois axes : la robustesse, la précision et finalement la comparaison des patrons de lumière non structurée aux autres méthodes.

Les patrons de lumière non structurée se différencient en premier lieu par leur robustesse aux interréflexions et aux discontinuités de profondeur. Ils sont conçus de sorte à homogénéiser la quantité d'illumination indirecte causée par la projection sur des surfaces difficiles. En contrepartie, la mise en correspondance des images projetées et capturées est plus complexe qu'avec les méthodes dites structurées. Une méthode d'appariement probabiliste et efficace est proposée afin de résoudre ce problème.

Un autre aspect important des reconstructions basées sur la lumière non structurée est la capacité de retrouver des correspondances sous-pixels, c'est-à-dire à un niveau de précision plus fin que le pixel. Nous présentons une méthode de génération de code de très grande longueur à partir des motifs de lumière non structurée. Ces codes ont l'avantage double de permettre l'extraction de correspondances plus précises tout en requérant l'utilisation de moins d'images. Cette contribution place notre méthode parmi les meilleures au niveau de la précision tout en garantissant une très bonne robustesse.

Finalement, la dernière partie de cette thèse s'intéresse à la comparaison des méthodes existantes, en particulier sur la relation entre la quantité d'images projetées

et la qualité de la reconstruction. Bien que certaines méthodes nécessitent un nombre constant d'images, d'autres, comme la nôtre, peuvent se contenter d'en utiliser moins aux dépens d'une qualité moindre. Nous proposons une méthode simple pour établir une correspondance *optimale* pouvant servir de référence à des fins de comparaison. Enfin, nous présentons des méthodes hybrides qui donnent de très bons résultats avec peu d'images.

Mots clés— vision 3D, vision par ordinateur, reconstruction active, lumière codée, lumière structurée, lumière non structurée, projecteur, caméra

ABSTRACT

This thesis deals with active 3D reconstruction from camera-projector systems. Standard reconstruction methods use coded light patterns that come with their strengths and weaknesses. We introduce unstructured light patterns that feature several improvements compared to the current state of the art. The research presented revolves around three main axes : robustness, precision and comparison of existing unstructured light patterns to existing methods.

Unstructured light patterns stand out first and foremost by their robustness to interreflections and depth discontinuities. They are specifically designed to homogenize the indirect lighting generated by their projection on hard to scan surfaces. The downside of these patterns is that matching projected and captured images is not straightforward anymore. A probabilistic correspondence method is formulated to solve this problem efficiently.

Another important aspect of reconstruction obtained with unstructured light patterns is their ability to recover subpixel correspondences, that is with a precision finer than the pixel level. We present a method to produce long codes using unstructured light. These codes enable us to extract more precise correspondences while requiring less patterns. This contribution makes our method one of the most accurate - yet robust to standard challenges - method of active reconstruction in the domain.

Finally, the last part of this thesis adresses the comparison of existing reconstruction methods on several aspects, but mainly on the impact of using less and less patterns on the quality of the reconstruction. While some methods need a fixed number of images, some, like ours, can accommodate fewer patterns in exchange for some quality loss. We devise a simple method to capture an *optimal* correspondence

map that can be used as a groundtruth for comparison purposes. Last, we present several hybrid methods that perform quite well even with few images.

Keywords— 3D vision, computer vision, active reconstruction, coded light, structured light, unstructured light, projector, camera

TABLE DES MATIÈRES

Table des Matières **i**

Liste des Figures **iv**

Liste des Tables **vii**

Introduction 1

Notions préliminaires 5

Chapitre 1 : Éléments de base en vision par ordinateur **5**

1.1 Modélisation de l'image 5

1.2 Notation et transformations 7

1.3 Géométrie des caméras et projecteurs 9

Chapitre 2 : Méthodes de reconstruction active **18**

2.1 Méthodes non temporelles 21

2.2 Méthodes temporelles 24

I Robustesse des méthodes de reconstruction active 31

Chapitre 3 : Les principaux défis de la lumière codée **32**

3.1 Propriétés photométriques de la caméra et du projecteur 32

3.2 Propriétés photométriques de la scène 34

Chapitre 4 : Unstructured Light Scanning Robust to Indirect Illumination and Depth Discontinuities (Article) 36

Abstract 37

4.1 Introduction 37

4.2 Previous work 39

4.3 Problems of structured light systems 41

4.4 Unstructured light patterns 45

4.5 Establishing pixel correspondence 49

4.6 Comparison with the Gupta *et al.* method 60

4.7 Experiments 61

4.8 Conclusion 66

II Précision des méthodes de reconstruction active 71

Chapitre 5 : Mise en correspondance sous-pixel 72

5.1 Ratio de pixels projecteur-caméra 72

5.2 Nécessité des correspondances sous-pixels 74

5.3 Estimation sous-pixel de la correspondance 74

Chapitre 6 : Subpixel Scanning Invariant to Indirect Lighting using Quadratic Code Length (Article) 79

Abstract 80

6.1 Introduction 80

6.2 Previous work 82

6.3 From linear to quadratic code length 83

6.4 Achieving subpixel accuracy 86

6.5 Experiments 90

6.6 Conclusion 100

III Comparaison des méthodes de reconstruction active 101

Chapitre 7 : Motivations d'une étude comparative des méthodes de lumière codée 102

7.1 Dualité quantité-qualité 103

7.2 Calcul de correspondances optimales 105

Chapitre 8 : A Comparison of Coded Light Methods for Precise and Robust Active Reconstruction (Article) 109

Abstract 109

8.1 Introduction 110

8.2 Previous work and state of the art 111

8.3 Experimental setup 114

8.4 Evaluation methodology 119

8.5 Experiments 125

8.6 Discussion and future work 131

Conclusion 133

Références 135

LISTE DES FIGURES

1.1 Caméra sténopé 6
1.2 Point principal d'un projecteur 13
1.3 Géométrie épipolaire 16

2.1 Reconstruction active à l'aide de lumière codée 19
2.2 Exemples de motifs pour des méthodes spatiales avec génération non formelle 22
2.3 Exemples de motifs pour des méthodes spatiales avec génération basée sur des algorithmes déterministes 23
2.4 Exemples de motifs utilisés par des méthodes temporelles 30

3.1 Composante de la radiance d'un point de la scène 34

4.1 Example of a scene and its correspondence maps 38
4.2 Incorrect pixel classification caused by interreflections 42
4.3 Illumination contribution for selected pixels 44
4.4 Example of generated noise patterns 46
4.5 Measure of code uniqueness for varying frequencies 47
4.6 Measure of local correlation for varying frequencies 47
4.7 Matching heuristics 55
4.8 Match convergence with and without use of heuristics 55
4.9 Typical histogram of matching costs and standard deviation of intensity 56
4.10 2D log-histograms of matching costs and standard deviations of intensity 57
4.11 Effects of using higher frequency on indirect lighting 59

4.12 Average correspondence cost as a function of pattern frequency for various code lengths 59
4.13 The four scenes used in experiments 62
4.14 Triangulation from the correspondences given by our method 64
4.15 Results for the `Ball` scene 67
4.16 Results for the `Games` scene 68
4.17 Results for the `Grapes & Peppers` scene 69
4.18 Results for the `Corner` scene 70

5.1 Différents ratios de pixels entre la caméra et le projecteur 73
5.2 Reconstruction avec et sans sous-pixel d'une scène 75
5.3 Incertitude d'une reconstruction non sous-pixel 76

6.1 A band-pass gray level pattern and a reconstruction 81
6.2 Computation of quadratic codes 84
6.3 2D zero-crossings provides constraints on the subpixel 87
6.4 RMS subpixel error as a function of blur and noise 90
6.5 Effects of gamma nonlinearity and number of patterns on the subpixel accuracy 91
6.6 The robot scene with and without artificial indirect lighting 92
6.7 Comparison of reconstructions for a X-Z projection of the robot . . . 93
6.8 Histogram of reconstruction variations on the robot scene 94
6.9 A complex scene comprised of several challenges 95
6.10 Reconstruction of the complex scene with two methods 96
6.11 Close-up of the reconstruction of a translucent ball 97
6.12 Close-up of the reconstruction of a corner between two walls 98
6.13 Correspondences of a thin object 99

7.1 Alignement des motifs projetés sur la géométrie épipolaire 104
7.2 Matrice d'illumination 107

8.1 Camera-projector calibration setup 115
8.2 Patterns used for the hybrid methods 119
8.3 Correspondence map acquired with our groundtruth method 122
8.4 The scene used in our experiments 123
8.5 Mask used to separate indirectly lit from valid pixels 124
8.6 Correspondence maps for the scene with the plane occluder 126
8.7 Difference with groundtruth for cropped regions of interest 127
8.8 Difference with groundtruth for cropped regions of interest 128
8.9 Correspondence maps for the scene with the bag occluder 130
8.10 Evolution of the subpixel correspondences as a function of the number of patterns used 132

LISTE DES TABLES

1.1 Hiérarchie des transformations 2D : le $+\cdots$ indique qu'une transformation préserve tous les invariants des transformations situées plus bas qu'elle. Tirée de [65](traduction libre). 8

REMERCIEMENTS

Les premiers remerciements de cette thèse vont à mon directeur de recherche, Sébastien Roy, pour m'avoir donné l'opportunité de rejoindre le laboratoire de vision 3D bien avant que je commence ma maîtrise. La liberté dans le choix du sujet de recherche m'a permis de toucher à tous les domaines de la vision par ordinateur, et bien que je m'y sois perdu en chemin, cela a fini par porter ses fruits.

Je tiens aussi à remercier tous les membres du laboratoire pour leurs discussion, collaboration et soutien. Une attention particulière va à Jamil Drareni, avec qui j'ai essayé tant d'idées non fructueuses, mais aux combien formatrices. Certains travaux présentés dans cette thèse viennent d'une simple discussion, il y a de cela longtemps, et pour laquelle il se doit de recevoir le crédit.

Je me dois aussi de remercier tous mes amis qui ont supporté mes séances de disparition chronique les veilles de dates de remise. Un énorme merci à Fredéric Checkoury qui m'a toujours poussé à aller chercher le meilleur de moi-même et sans qui je n'aurai pas pu finir. Enfin, une mention spéciale à Florian Gabert pour sa créativité, son aide dans la réalisation de certaines figures et pour tout le reste aussi.

Finalement, tout ceci n'aurait pas pu être possible sans le soutien de ma famille qui depuis presque 10 ans me soutient à distance et me réchauffe le coeur à chaque visite. Merci à mes soeurs pour le soutien moral, à mes parents pour tout ce qu'ils m'apportent et pour leur confiance en moi depuis ma plus tendre enfance.

Ma dernière pensée va à ma chienne Laïka, qui m'a permis de ne pas sombrer dans la dépression par son enthousiasme débordant et son amour inconditionnel. Bien qu'elle ait beaucoup trop d'énergie pour rester en place plus que trois secondes et se laisser reconstruire en 3D pour la science, je lui dédie cette thèse pour toutes

les journées interminables passées à m'attendre devant la porte.

INTRODUCTION

La vision par ordinateur, comme certaines autres disciplines de l'informatique, se porte naturellement à des applications industrielles. Par exemple, les nouveaux formats HD d'images ont introduit une course à l'efficacité pour les traitements d'imagerie en temps réel. En retour, la vision par ordinateur tire aussi partie de l'émergence de nouvelles technologies pour améliorer ou introduire de nouveaux domaines de recherche. Ainsi, on peut penser à l'ajout d'algorithmes de suivi de visages dans nos appareils photo de tous les jours, ou la montée en puissance des systèmes de divertissement sans contrôleur physique qui utilisent des algorithmes de détection du corps humain comme outil d'interaction.

Dans ce contexte, les projecteurs ont été utilisés en vision par ordinateur depuis plusieurs années. Dans un premier temps, employés comme outils de présentation, l'essentiel de la recherche portait sur l'adaptation de la projection en fonction du support d'affichage. Une caméra était utilisée pour observer les images projetées et apporter des corrections géométriques et photométriques aux prochaines images envoyées par le projecteur. La réalité augmentée et les systèmes multi projecteurs sont des exemples d'applications de cette recherche. Par la suite, les projecteurs ont été utilisés comme des éléments actifs du système et ont donné naissance à plusieurs domaines de la vision. La stéréo augmentée, par exemple, utilise l'information fournie par un projecteur pour "ajouter" de la texture à une scène afin d'aider l'algorithme de mise en correspondance dans les zones trop uniformes. La reconstruction active à l'aide de lumière codée, est un autre domaine de recherche qui tire profit de l'information additionnelle provenant de la projection. Le terme *actif* vient de l'utilisation des projecteurs non pas comme outil de présentation, mais comme un élément per-

mettant de participer activement et d'aider la méthode.

Les travaux de cette thèse portent sur les méthodes de reconstruction active à l'aide de projecteurs. Par opposition aux méthodes de reconstruction passive où l'algorithme doit utiliser l'information disponible sans interagir avec la scène observée, les méthodes de reconstruction actives tentent de "générer" les données de façon optimale, afin de simplifier le travail de tout algorithme subséquent. Les projecteurs sont des outils particulièrement bien adaptés pour cette tâche, puisque l'image projetée peut être entièrement construite de sorte à assister un algorithme de reconstruction. Ainsi, les méthodes dites de *lumière codée* - un terme que nous utiliserons au profit du terme plus commun, mais incomplet de *lumière structurée* - ont été utilisées dans l'industrie depuis quelques décennies pour leur simplicité et performance.

Organisation de la thèse

Cette thèse est séparée en trois parties détaillant nos contributions au domaine de la vision par ordinateur, et en particulier aux méthodes de reconstruction active. Chaque partie sera composée de deux chapitres, un premier introduisant les concepts essentiels à la compréhension du second. Celui-ci est présenté sous forme d'un article soumis à une conférence ou à un journal scientifique.

Dans un premier temps, les notions de vision et les notations mathématiques nécessaires à la lecture de cette thèse seront introduites dans le chapitre 1. Le chapitre 2 propose un survol de l'état de l'art en ce qui concerne les méthodes de reconstruction active.

La première partie de cette thèse s'intéresse à la **robustesse** des méthodes de reconstruction active par lumière codée. Les principaux défis auxquels ces méthodes font face et en particulier, le problème de l'illumination indirecte sont présentés dans le chapitre 3. Le chapitre 4 propose les motifs de lumière non structurée comme solution au problème des interréflexions.

Il est question de la **précision** des reconstructions dans la seconde partie. Le cha-

pitre 5 introduit le concept de mise en correspondance sous-pixels, qui permet d'améliorer la précision d'une reconstruction. La méthode dite de combinaison quadratique des codes, s'appuyant sur la lumière non structurée, ajoute cette amélioration tout en utilisant moins de patrons. Cette méthode sera présentée au chapitre 6.

Finalement, la dernière partie synthétise et évalue les forces et faiblesses des méthodes connues, sous forme d'une **comparaison**. Le chapitre 7 fait état de la motivation derrière l'idée de cette évaluation. Différents sujets y sont exposés, telle la dualité qualité/quantité à laquelle toutes les méthodes de lumière codée font face, ainsi que la nécessité de comparer uniquement les correspondances dans l'image, et non les reconstructions 3D. Cette comparaison est exposée au chapitre 8, accompagnée d'une méthode pour produire une correspondance de référence à des fins de comparaison. Des méthodes hybrides donnant de bons résultats avec peu d'images sont également présentées.

En guise de conclusion, nous mettrons en perspective la portée de nos contributions au domaine de la vision par ordinateur et discuterons des possibilités de futurs travaux de recherche.

Notions préliminaires

Chapitre 1

ÉLÉMENTS DE BASE EN VISION PAR ORDINATEUR

1.1 Modélisation de l'image

Une caméra numérique telle que celles utilisées en vision par ordinateur est essentiellement un outil permettant de convertir un rayonnement électromagnétique (la lumière sous forme de photons) en un signal électrique analogique, amplifié et traité pour devenir une image numérique [72].

Il est très clair que modéliser la transformation entre une scène 3D et l'image 2D obtenue à partir de celle-ci à l'aide de modèles physiques tels que le précédent est inconcevable. En vision par ordinateur, une caméra (terme générique qui décrit tous les appareils de prise de vue : caméra film, vidéo, appareil photo numérique, ...) est un outil qui transforme les objets physiques 3D en image 2D. Cette transformation peut se décomposer en deux parties : **géométrique** et **photométrique**.

En imagerie, on utilise le terme **pixel** pour définir la plus petite unité de surface dans une image : un pixel est défini par sa position 2D et sa couleur. De même, dans la scène, à chaque unité de surface est associé un **voxel** : un pixel 3D. Pour passer d'un voxel à un pixel, la transformation géométrique, modélisée par la caméra sténopé [65, 29, 41] (voir figure 1.1), définit la position de chaque pixel en fonction d'un voxel et des paramètres de caméra. La transformation photométrique, quant à elle, utilise le modèle lambertien pour définir la couleur de chaque pixel en fonction d'un voxel et des propriétés de sa surface. Le modèle lambertien n'est pas toujours valide (e.g. surface spéculaire), mais c'est celui qui est utilisé en vision pour sa simplicité.

La suite de nos travaux étant beaucoup plus axée sur la géométrie des caméras et projecteurs, il est nécessaire de placer l'emphase sur l'aspect géométrique de la

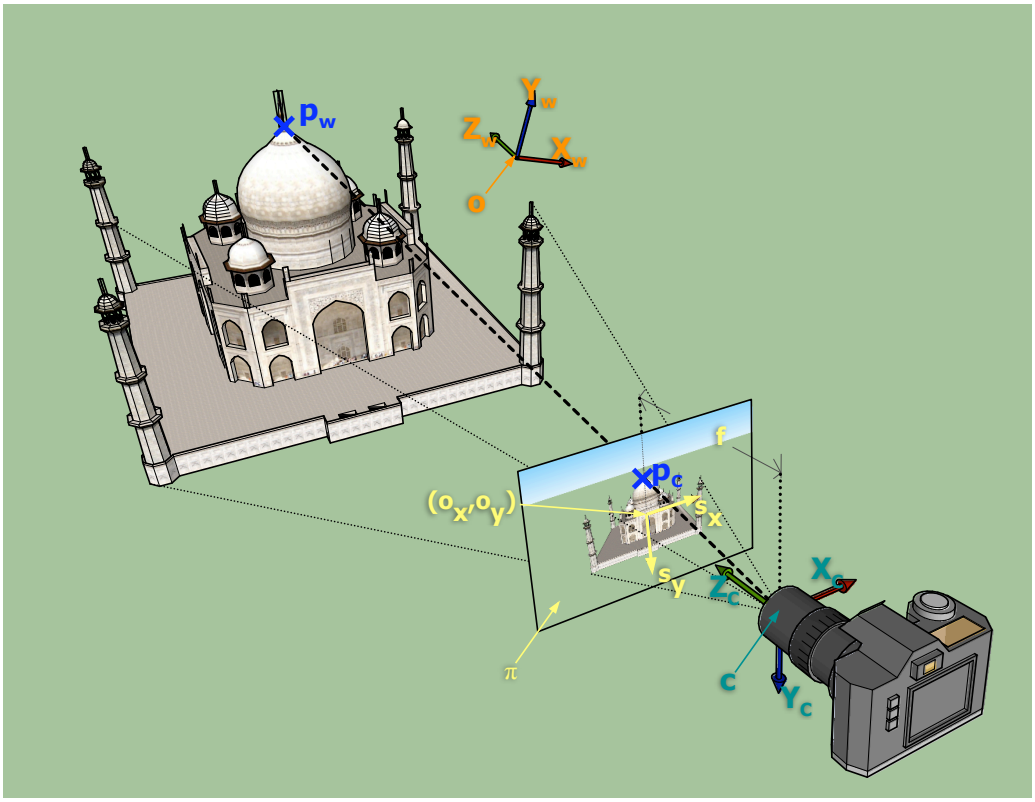


FIGURE 1.1: La caméra sténopé simplifie le modèle de formation de l'image en assurant que chaque rayon dans la scène se projette dans l'image en passant par un seul point : le centre optique [65].

formation de l'image. L'aspect photométrique s'applique indépendamment du modèle géométrique utilisé, et pour le reste de cette thèse, il est sous-entendu que le modèle lambertien est toujours utilisé pour déterminer la couleur d'un pixel. C'est pourquoi nous ne rentrerons pas dans les détails du modèle lambertien (une description détaillée est présentée dans [41, 65]). Les termes voxels et pixels seront souvent interchangés, respectivement avec point 3D et 2D.

1.2 Notation et transformations

Pour le reste du document, on exprimera les positions 2D ou 3D par des vecteurs. Un point 2D $\boldsymbol{p} \in \mathbb{R}^2$ est noté en gras, un point 3D $\boldsymbol{p}_w \in \mathbb{R}^3$ est aussi noté en gras, mais l'indice exprime souvent le référentiel dans lequel le point est exprimé (et donc sa dimensionnalité). Les indices w, c et i seront utilisés pour exprimer un point dans le monde (3D), dans la caméra (3D) et dans l'image de la caméra (2D). Les vecteurs sont par définition en format colonne, ainsi $\boldsymbol{v} \in \mathbb{R}^3$ a pour dimension 3×1, et $\boldsymbol{v}^{\mathrm{T}}$, sa transposée a pour dimension 1×3 et est un vecteur ligne.

Les matrices seront notées $\boldsymbol{M}$ en gras et en majuscule. De même, les indices des matrices indiqueront souvent leur signification. On représente la transposée d'une matrice par $\boldsymbol{M}^{\mathrm{T}}$, son inverse $\boldsymbol{M}^{-1}$ et sa transpose-inverse (ou inverse-transpose) $\boldsymbol{M}^{-\mathrm{T}}$. Lorsque la dimension n'est pas claire, on la précisera par la notation $\boldsymbol{M}_{(n \times m)}$ où n est le nombre de lignes, et m le nombre de colonnes. Plusieurs matrices sont standards, $\boldsymbol{E}$ est la matrice essentielle, $\boldsymbol{F}$ est la fondamentale, $\boldsymbol{H}$ représente une homographie, $\boldsymbol{I}$ est la matrice identité. La matrice $[\boldsymbol{v}]_{\times}$ est la matrice issue du produit vectoriel de $\boldsymbol{v} \in \mathbb{R}^3$ avec n'importe quel autre vecteur $\boldsymbol{u} \in \mathbb{R}^3$. En effet, $\boldsymbol{v} \times \boldsymbol{u} = [\boldsymbol{v}]_{\times(3 \times 3)}\boldsymbol{u}$.

Finalement, on définira un plan $\boldsymbol{\pi}$ par son équation $\boldsymbol{n}^{\mathrm{T}}\boldsymbol{p} + d = 0$, valide pour tout point $\boldsymbol{p} \in \mathbb{R}^3$ sur le plan. $\boldsymbol{n}$ est la normale du plan, et d est sa distance à l'origine. On utilise souvent la notation compacte $\boldsymbol{\pi} = \left(\boldsymbol{n}^{\mathrm{T}}, d\right)^{\mathrm{T}}$.

Les transformations entre systèmes de coordonnées du plan sont très importantes en vision, et sont représentées par des matrices, ce sont des transformations linéaires. On considère la rotation, translation, mise à l'échelle ; les transformations affines, projectives ... Chacune de ces transformations conserve un certain nombre de propriétés géométriques. Elles suivent une hiérarchie de conservation des invariants qui est montrée à la table 1.1[65].

La transformation la plus générale, la transformation projective (appelée homographie) est aussi la plus utilisée en vision. Elle décrit n'importe quelle transfor-

Nom	Matrice	# Degrés de liberté	Préserve :	Symbole
translation	$\left[\boldsymbol{I}\middle\vert\boldsymbol{t}\right]_{(2\times 3)}$	2	orientation+$\cdots$	
euclidienne	$\left[\boldsymbol{R}\middle\vert\boldsymbol{t}\right]_{(2\times 3)}$	3	longueurs+$\cdots$	
similarité	$\left[s\boldsymbol{R}\middle\vert\boldsymbol{t}\right]_{(2\times 3)}$	4	angles+$\cdots$	
affine	$\boldsymbol{A}_{(2\times 3)}$	6	parallélisme +$\cdots$	
projective	$\boldsymbol{H}_{(3\times 3)}$	8	lignes droites +$\cdots$	

TABLE 1.1: Hiérarchie des transformations 2D : le $+\cdots$ indique qu'une transformation préserve tous les invariants des transformations situées plus bas qu'elle. Tirée de [65](traduction libre).

mation d'un plan à un autre. Pour la représenter, on utilise une matrice $\boldsymbol{H}_{(3\times 3)}$ qui n'opère pas dans l'espace euclidien, mais dans l'espace projectif $\mathbb{P}^2$. Les coordonnées projectives du point 2D $\boldsymbol{p} = (x,y)^{\mathrm{T}}$, exprimées dans cet espace sont notées $\tilde{\boldsymbol{p}} \in \mathbb{P}^2 = (x',y',w)^{\mathrm{T}}$. L'introduction d'une dimension supplémentaire permet de représenter les entités à l'infini dans le cas particulier où $w = 0$ (les homographies peuvent projeter des points à l'infini...). En effet, le passage de $\mathbb{P}^2$ à $\mathbb{R}^2$ s'effectue par la manipulation suivante :

$$\boldsymbol{p} = (x'/w, y'/w)^{\mathrm{T}} \tag{1.1}$$

(si $w = 0$ alors $x'/w = y'/w = \infty$). De manière générale, pour faire passer un point d'un espace euclidien à un espace projectif, on peut choisir n'importe quel facteur d'échelle $w \neq 0$ (on choisit souvent $w = 1$ par simplicité).

Les égalités dans $\mathbb{P}^2$ (et $\mathbb{P}^3$) sont définies à un facteur d'échelle près et sont notées par $\tilde{\boldsymbol{p}} \propto \tilde{\boldsymbol{q}}$, qui signifie que $\tilde{\boldsymbol{p}}$ est projectivement équivalent à $\tilde{\boldsymbol{q}}$. Cela s'écrit aussi $\tilde{\boldsymbol{p}} \times \tilde{\boldsymbol{q}} = \mathbf{0}$. En projetant $\tilde{\boldsymbol{p}}$ et $\tilde{\boldsymbol{q}}$ sur $\mathbb{R}^2$ en utilisant (1.1), l'égalité devient stricte (elle n'est plus définie à un facteur d'échelle).

1.3 Géométrie des caméras et projecteurs

Cette section définit les équations de base dérivées à partir du modèle sténopé montré à la figure 1.1. Nous verrons qu'elles s'appliquent aussi bien à la caméra qu'au projecteur.

1.3.1 Géométrie d'une caméra

La **caméra sténopé** est composée de deux éléments principaux : un **centre optique** $\boldsymbol{c}$ (qui modélise l'objectif de nos caméras actuelles) et un **plan image** ou plan de projection $\boldsymbol{\pi}$ (qui modélise le CCD ou CMOS : le senseur d'une caméra). Un point 3D $\boldsymbol{p}_w \in \mathbb{R}^3$ est projeté dans l'image au point 2D $\boldsymbol{p} \in \mathbb{R}^2$ par une transformation projective 3D $\boldsymbol{M}_{(3\times 4)}$ appelée **matrice de caméra**. Cette transformation décrit le passage de tout point exprimé dans le référentiel du monde à sa projection dans l'image. Cette transformation se décompose en trois étapes : passage du repère du monde à celui de la caméra par la transformation rigide $\boldsymbol{M}_{w(4\times 4)}$, passage du repère de la caméra à celui du plan image par la transformation projective $\boldsymbol{M}_{c(3\times 4)}$ et finalement passage du repère image au repère pixel par la transformation affine $\boldsymbol{M}_{p(3\times 3)}$.

La matrice $\boldsymbol{M}_w$ est une transformation rigide qui peut se décomposer en une rotation $\boldsymbol{R}$ qui aligne les axes du repère du monde (X_w,Y_w,Z_w) sur ceux du repère de la caméra (X_c,Y_c,Z_c), et une translation $\boldsymbol{t}$ qui déplace l'origine du monde $\boldsymbol{o}$ sur le centre de la caméra $\boldsymbol{c}$. Le point $\boldsymbol{p}_w$ dans le monde est relié au point $\boldsymbol{p}_c$ dans la

caméra par $\boldsymbol{p}_c = \boldsymbol{R}\boldsymbol{p}_w + \boldsymbol{t}$. En d'autres termes :

$$\tilde{\boldsymbol{p}}_c \propto \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \tilde{\boldsymbol{p}}_w \propto \boldsymbol{M}_w \tilde{\boldsymbol{p}}_w,$$

où $\tilde{\boldsymbol{p}}_c$ et $\tilde{\boldsymbol{p}}_w$ représentent $\boldsymbol{p}_c$ et $\boldsymbol{p}_h$ en notation homogène.

La matrice $\boldsymbol{M}_c$ exprime le passage du repère de la caméra à celui de l'image. Le point $\boldsymbol{p}_c$ est relié au point $\boldsymbol{p}_i$ sur $\boldsymbol{\pi}$ par

$$\tilde{\boldsymbol{p}}_i \propto \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tilde{\boldsymbol{p}}_c \propto \boldsymbol{M}_c \tilde{\boldsymbol{p}}_c,$$

où $\tilde{\boldsymbol{p}}_i$ est en coordonnées homogènes. C'est ici que s'effectue "une perte d'information", puisqu'une dimension disparaît lors de cette étape (la dernière colonne de $\boldsymbol{M}_c$ est nulle). Cette transformation n'est donc pas inversible : on ne peut pas aller du repère de l'image à celui de la caméra (sans informations supplémentaires).

La matrice $\boldsymbol{M}_p$ exprime le passage du repère image au repère pixel : il s'agit d'une transformation affine qui ajuste l'origine et l'échelle des coordonnées. Le point $\boldsymbol{p}_i$ sur $\boldsymbol{\pi}$ est relié au point $\boldsymbol{p}$ dans l'image (en pixels) par :

$$\tilde{\boldsymbol{p}} \propto \begin{bmatrix} s_x & 0 & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \tilde{\boldsymbol{p}}_i \propto \boldsymbol{M}_p \tilde{\boldsymbol{p}}_i,$$

où $\tilde{\boldsymbol{p}}$ est la notation homogène de $\boldsymbol{p}$. s_x et s_y expriment la taille d'un pixel (en fonction de l'unité du plan image : le senseur) le long des dimensions x et y, alors que (o_x, o_y) est la coordonnée du **point principal** (le point d'intersection entre l'**axe optique** et le plan image, l'axe optique étant la droite perpendiculaire au plan image passant par le centre de la caméra). Cette transformation permet de passer des unités physiques (millimètre par exemple) aux unités pixels (grâce à la mise à l'échelle), et d'ajuster l'origine du repère pixel (en vision, l'origine est placée au coin en haut-gauche et non pas au centre).

Il faut noter que le modèle sténopé traditionnel projette l'image sur un plan image placé en **z**=**-f**, c'est-à-dire "derrière" la caméra. L'image ainsi obtenue est inversée verticalement, et on la "retourne" pour obtenir la vraie projection. L'effet de cette inversion est le même que de considérer que le plan image est placé en **z**=**f**, et d'inverser le sens de l'axe des y. Dans la figure 1.1, nous avons volontairement représenté le plan image en avant de la caméra, et c'est pourquoi le système d'axe de la caméra utilise un "système main gauche", qui est le système standard en vision par ordinateur.

Finalement, on peut écrire sous forme de produits matriciels la transformation du point $\boldsymbol{p}_w$ dans le monde, vers le point $\boldsymbol{p}$ dans l'image, en notation homogène :

$$\begin{aligned}
\tilde{\boldsymbol{p}} &\propto \begin{bmatrix} s_x & 0 & o_x \\ 0 & s_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0} & 1 \end{bmatrix} \tilde{\boldsymbol{p}}_w \\
&\propto \begin{bmatrix} fs_x & 0 & o_x \\ 0 & fs_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \tilde{\boldsymbol{p}}_w \\
\tilde{\boldsymbol{p}} &\propto \boldsymbol{K} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \tilde{\boldsymbol{p}}_w. \qquad (1.2)
\end{aligned}$$

Cette dernière forme est celle que nous utiliserons tout au long de cette thèse. Elle encapsule dans $\boldsymbol{K}_{(3\times 3)}$ la matrice des **paramètres internes**, et dans $[\boldsymbol{R} \mid \boldsymbol{t}]_{(3\times 4)}$ la matrice des **paramètres externes** [29]. Par conséquent : $\boldsymbol{M} = \boldsymbol{K}[\boldsymbol{R} \mid \boldsymbol{t}]$. On utilisera aussi sa notation homogène $\tilde{\boldsymbol{M}} = \begin{bmatrix} \boldsymbol{K} & \boldsymbol{0} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}^{\mathrm{T}} & 1 \end{bmatrix}$, qui est inversible, et permet donc de déprojeter un point de l'image vers un point 3D à un facteur d'échelle près (sa profondeur qui est inconnue).

Les paramètres internes représentent les caractéristiques invariantes d'une caméra (seul le zoom affecte ces paramètres). Si le zoom reste fixe, il n'est nécessaire de les estimer qu'une seule fois. Les paramètres externes représentent quant à eux la pose de la caméra par rapport au centre du monde. Si la caméra se déplace, ces paramètres

vont varier.

Les paramètres internes doivent parfois tenir compte d'une distorsion qui ne peut être modélisée par une fonction linéaire. La **distorsion radiale** modifie l'image de sorte que les lignes droites ne passant pas par le point principal apparaissent d'autant plus courbes qu'elles sont proches du centre de distorsion, souvent le centre de l'image [65]. Cette transformation s'effectue dans le système de caméra normalisé, c'est-à-dire après projection depuis le repère de la caméra vers le plan image normalisé $\boldsymbol{\pi}_0$ d'équation $\mathbf{z}=\mathbf{1}$ (par opposition à $\mathbf{z}=\mathbf{f}$ pour le plan image).

La projection du point $\boldsymbol{p}_c$ sur $\boldsymbol{\pi}_0$ sera notée $\hat{\boldsymbol{p}}_c$ (en coordonnées homogènes) et peut s'obtenir en utilisant :

$$\hat{\boldsymbol{p}}_c \propto \begin{bmatrix} \boldsymbol{I}_{(3\times 3)} & \mathbf{0} \end{bmatrix} \tilde{\boldsymbol{p}}_c,$$

ou bien depuis les coordonnées images $\boldsymbol{p}$:

$$\hat{\boldsymbol{p}}_c \propto \boldsymbol{K}^{-1}\tilde{\boldsymbol{p}}.$$

En fixant $\hat{\boldsymbol{p}}_c = (x', y', 1)$, on peut calculer les coordonnées distorsionnées $\boldsymbol{p}_c^* = (x^*, y^*)$ avec le modèle suivant :

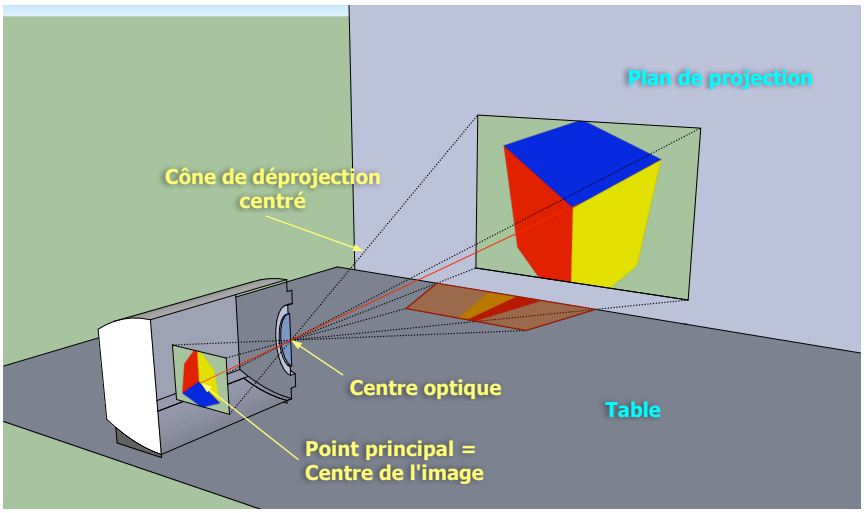
$$\begin{aligned} x^* &= x'(1 + k_1 r'^2 + k_2 r'^4) \\ y^* &= y'(1 + k_1 r'^2 + k_2 r'^4), \end{aligned} \tag{1.3}$$

avec $r'^2 = x'^2 + y'^2$ et (k_1, k_2) les coefficients de distorsion. Bien sûr, on peut ensuite obtenir les coordonnées images en utilisant $\tilde{\boldsymbol{p}}^* = \boldsymbol{K}\tilde{\boldsymbol{p}}_c^*$.

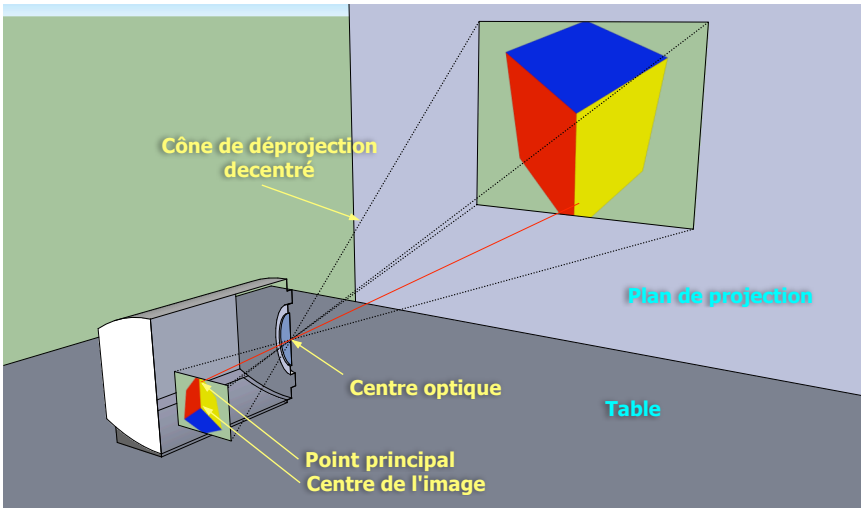
1.3.2 Géométrie d'un projecteur

Si une caméra permet la capture d'image, le projecteur, lui, en permet la projection. Complètement différents du point de vue du fonctionnement, ils sont néanmoins tous les deux modélisés à l'aide du modèle sténopé. La différence entre les deux est uniquement conceptuelle ; en ce sens que les rayons reliant les points du monde à

ceux de l'image, s'ils devaient avoir un sens, seraient inversés : ils seraient sortants plutôt que rentrants.



(a)



(b)

FIGURE 1.2: Le cône de déprojection d'un projecteur n'est pas centré sur l'axe optique pour permettre la projection à plat sur une table. La zone en rouge sur la table dans (a) n'apparaît pas lorsque le cône de déprojection est désaxé (b).

Bien que le modèle sténopé s'applique aussi bien à la caméra qu'au projecteur, il

est à noter que la distorsion radiale affecte peu ou pas les images projetées par un projecteur. Il est rarement nécessaire de la prendre en compte. Une autre différence est que le point principal d'un projecteur ne correspond jamais avec le centre de l'image. Puisque le projecteur est généralement construit pour être posé à plat sur une table, il est nécessaire que le **cône de déprojection** soit désaxé (comme à la figure 1.2) : il n'est pas centré sur l'axe optique (autrement une grosse partie de l'image se retrouverait projetée sur la table, comme c'est le cas dans la figure).

1.3.3 Géométrie d'un système caméra-projecteur

Déprojection-reprojection

Soient $\boldsymbol{M}_c$ et $\boldsymbol{M}_p$, les matrices de projection de la caméra et du projecteur. Pour rappel, $\boldsymbol{M}_c = \boldsymbol{K}_c\,[\boldsymbol{R}_c \mid \boldsymbol{t}_c]$, et $\boldsymbol{M}_p = \boldsymbol{K}_p\,[\boldsymbol{R}_p \mid \boldsymbol{t}_p]$. Nous avons vu qu'un point $\boldsymbol{p}_w = (X,Y,Z)^{\mathrm{T}}$ se projette dans l'image de la caméra en $\boldsymbol{p} = (x,y)^{\mathrm{T}}$ et dans l'image du projecteur en $\boldsymbol{q}$ grâce à l'équation de projection (1.2). On peut inversement définir l'équation de déprojection qui déprojette $\boldsymbol{p}$ vers $\boldsymbol{p}_w$ à l'aide de $\tilde{\boldsymbol{M}}_c$:

$$\tilde{\boldsymbol{p}}_w \propto \tilde{\boldsymbol{M}}_c^{-1} \begin{pmatrix} x \\ y \\ 1 \\ d \end{pmatrix} \propto \tilde{\boldsymbol{M}}_c^{-1} \big(p^{\mathrm{T}},1,d\big)^{\mathrm{T}},$$

où d représente la disparité d'un pixel (on remarque que si la caméra est en position canonique, i.e. si $\boldsymbol{R}_c = \boldsymbol{I}$ et $\boldsymbol{t}_c = \boldsymbol{0}$, alors $d = 1/Z$). On peut réappliquer l'équation (1.2) pour obtenir :

$$\tilde{\boldsymbol{q}} \propto \boldsymbol{M}_p \tilde{\boldsymbol{M}}_c^{-1} \big(p^{\mathrm{T}},1,d\big)^{\mathrm{T}}. \tag{1.4}$$

En principe, on ne connaît pas d (puisqu'elle dépend de Z qui est inconnu avec une seule image). Cette méthode est néanmoins utile lorsque l'on veut essayer plusieurs valeurs de disparité. En stéréo, par exemple, le but est de retrouver la profondeur d'un

point, et donc il est commun d'essayer plusieurs valeurs de disparité pour extraire celle qui semble la plus probable.

Géométrie épipolaire

La géométrie épipolaire décrit la relation entre deux points (issus de la projection du même point 3D) pris dans deux images de la même scène vue de deux points de vue différents. Elle est exprimée par une matrice qui revêt deux formes selon que les paramètres internes des deux caméras (ou caméra-projecteur) sont connus ou non.

Soit $\boldsymbol{p}_w$ un point du monde, et $\boldsymbol{p}_1$ et $\boldsymbol{p}_2$ sa projection dans les deux images. Il est clair depuis la figure 1.3 que $\boldsymbol{p}_w$, $\boldsymbol{p}_1$, $\boldsymbol{p}_2$ et les centres de la caméra et du projecteur sont coplanaires. Ce plan est appelé **plan épipolaire**. La contrainte de coplanarité est appelée **contrainte épipolaire**, et elle a plusieurs implications. La première est que le point correspondant à $\boldsymbol{p}_1$ dans l'image correspondante n'a d'autres choix que d'être à l'intersection du plan épipolaire associée et du plan image du projecteur. Cette intersection est une ligne dans l'image appelée **ligne épipolaire**. Ainsi, la contrainte épipolaire énonce qu'il n'est pas nécessaire de chercher dans toute l'image pour trouver une correspondance, il suffit de chercher le long de la ligne épipolaire associée. La projection du centre d'une des caméras dans l'autre image est appelée **épipole**. $\boldsymbol{e}_1$ est l'épipole associé à $\boldsymbol{c}_1$, le centre optique de la caméra. La ligne joignant les deux centres est appelée "**baseline**". Un plan est épipolaire si et seulement si il contient le "baseline". Un deuxième plan épipolaire figure dans la figure, $\boldsymbol{o}_1$ et $\boldsymbol{o}_2$ sont les images de $\boldsymbol{o}_w$, et toutes les propriétés énoncées plus haut s'appliquent aussi pour le plan défini par ces trois points.

On cherche la rotation $\boldsymbol{R}$ et la translation $\boldsymbol{t}$ reliant la caméra au projecteur. Puisque nous connaissons les paramètres internes de chaque acteur, nous travaillerons avec les coordonnées normalisées $\hat{\boldsymbol{p}}_1 = \boldsymbol{K}_c^{-1}\tilde{\boldsymbol{p}}_1$ et $\hat{\boldsymbol{p}}_2 = \boldsymbol{K}_p^{-1}\tilde{\boldsymbol{p}}_2$. On peut, sans perte de généralité, aligner la caméra avec le monde (elle est alors en position canonique), alors $\boldsymbol{M}_c = \left[\boldsymbol{I}_{(3\times 3)} \mid \boldsymbol{0}\right]$ et $\boldsymbol{M}_p = [\boldsymbol{R} \mid \boldsymbol{t}]$. L'épipole associé au centre de la caméra

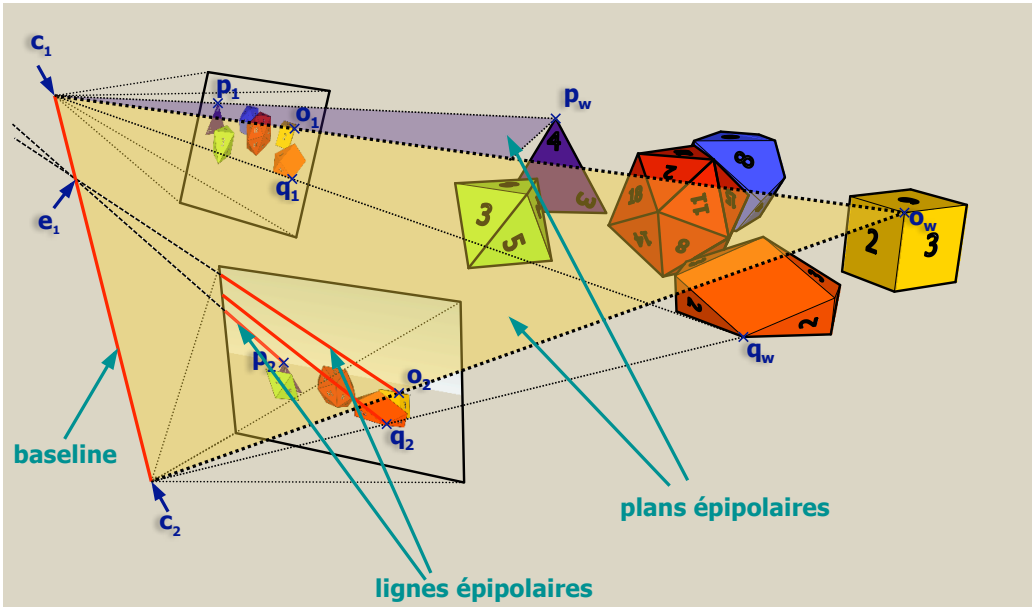


FIGURE 1.3: Les points $\boldsymbol{p}_w$, $\boldsymbol{p}_1$ et $\boldsymbol{p}_2$ sont sur un plan épipolaire. La ligne d'intersection entre un plan épipolaire et le plan image est la ligne épipolaire. Il y a une ligne associée à $\boldsymbol{p}_1$, et une associée à $\boldsymbol{p}_2$. Il en va de même pour le triplet de points $\boldsymbol{o}_w$, $\boldsymbol{o}_1$ et $\boldsymbol{o}_2$.

(le vecteur $\mathbf{0}$) est :

$$\boldsymbol{e}_1 = \boldsymbol{M}_p \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} = \boldsymbol{t}.$$

Pour définir la ligne épipolaire associée au point $\hat{\boldsymbol{p}}_1$, on a besoin de deux points. L'un d'entre eux est facile, l'épipole associé au centre de la caméra : $\boldsymbol{e}_1$. L'autre est issu de la déprojection de $\hat{\boldsymbol{p}}_1$. Puisqu'on ne connaît pas sa profondeur, on va le déprojeter sur le plan à l'infini $\boldsymbol{\pi}_\infty$: l'équation (1.4) avec $d = 0$ (équivaut à une profondeur infinie) devient :

$$\boldsymbol{M}_p \tilde{\boldsymbol{M}}_c^{-1} \begin{pmatrix} \hat{\boldsymbol{p}}_1 \\ 0 \end{pmatrix} = [\boldsymbol{R} \mid \boldsymbol{t}]\, \boldsymbol{I}_{(4\times 4)} \begin{pmatrix} \hat{\boldsymbol{p}}_1 \\ 0 \end{pmatrix} = \boldsymbol{R}\hat{\boldsymbol{p}}_1.$$

Avec deux points, on peut définir $\boldsymbol{l}_1$ la ligne épipolaire associée à $\hat{\boldsymbol{p}}_1$: $l_1 \propto \boldsymbol{e}_1 \times \boldsymbol{R}\hat{\boldsymbol{p}}_1 =$

$[\boldsymbol{t}]_\times \boldsymbol{R}\hat{\boldsymbol{p}}_1$. Et finalement, puisque $\hat{\boldsymbol{p}}_2$ est sur $\boldsymbol{l}_1$, on a :

$$\hat{\boldsymbol{p}}_2 \cdot \boldsymbol{l}_1 = 0 \Leftrightarrow \hat{\boldsymbol{p}}_2^{\mathrm{T}} [\boldsymbol{t}]_\times \boldsymbol{R}\hat{\boldsymbol{p}}_1 = 0 \Leftrightarrow \hat{\boldsymbol{p}}_2^{\mathrm{T}} \boldsymbol{E}_{(3\times 3)} \hat{\boldsymbol{p}}_1 = 0. \tag{1.5}$$

L'équation (1.5) est la relation de la géométrie épipolaire qui pour un point d'une image, détermine l'équation de sa ligne épipolaire, sur laquelle se situe le point correspondant dans l'image associée. La matrice $\boldsymbol{E} = [\boldsymbol{t}]_\times \boldsymbol{R}$ est la **matrice essentielle**. Cette matrice peut être calculée, en connaissant les paramètres internes des deux caméras et la correspondance d'au minimum 5 points [29]. En principe, on en choisit beaucoup plus afin d'en obtenir une estimation robuste. Si l'on ne connaît pas les paramètres internes, on peut toujours estimer la géométrie épipolaire à l'aide de la **matrice fondamentale**. La relation (1.5) devient simplement :

$$\hat{\boldsymbol{p}}_2^{\mathrm{T}} \boldsymbol{E}_{(3\times 3)} \hat{\boldsymbol{p}}_1 = 0 \Leftrightarrow \left(\boldsymbol{K}_p{}^{-1} \tilde{\boldsymbol{p}}_2\right)^{\mathrm{T}} \boldsymbol{E}_{(3\times 3)} \left(\boldsymbol{K}_c{}^{-1} \tilde{\boldsymbol{p}}_1\right) = 0 \Leftrightarrow \tilde{\boldsymbol{p}}_2^{\mathrm{T}} \boldsymbol{F}_{(3\times 3)} \tilde{\boldsymbol{p}}_1 = 0, \tag{1.6}$$

avec $\boldsymbol{F} = \boldsymbol{K}_p{}^{-\mathrm{T}} \boldsymbol{E}_{(3\times 3)} \boldsymbol{K}_c{}^{-1}$. Ces matrices seront utiles pour déterminer le mouvement entre deux caméras. En effet, il est possible de factoriser $\boldsymbol{E}$ (ou $\boldsymbol{F}$) de manière à retrouver $\boldsymbol{R}$ et $\boldsymbol{t}$. La démonstration est trop longue pour être détaillée ici, elle est disponible dans [29].

Chapitre 2

MÉTHODES DE RECONSTRUCTION ACTIVE

La mise en correspondance de deux images est un problème difficile pour plusieurs raisons. Premièrement, il n'existe pas de métrique magique pour décider si deux pixels sont correspondants : la couleur est souvent utilisée, mais peut s'avérer insuffisante, surtout dans les zones uniformes (e.g un mur blanc). D'autre part, la recherche de correspondances se fait dans un espace de grandes dimensions. Sans informations supplémentaires, une méthode brute doit examiner chaque pixel de l'image associée afin d'en trouver le plus similaire.

Le terme **reconstruction active** fait référence à toute méthode qui "produit" de l'information pour aider ou améliorer les performances de l'algorithme d'appariement. L'utilisation d'un projecteur afin "d'ajouter de l'information" à la scène permet de simplifier la mise en correspondance. En effet, la projection peut être vue comme une texture, qui impose directement des contraintes sur la métrique à utiliser. De plus, l'espace de recherche peut aussi être réduit puisque la projection impose des contraintes géométriques sur les correspondances. Le projecteur est donc une entité active de l'algorithme, qui permet de résoudre efficacement ce problème. La figure 2.1 illustre le processus de reconstruction active à l'aide de lumière codée. Les contributions de nos travaux permettent d'améliorer l'étape de mise en correspondance qui se situe entre la projection/acquisition et la triangulation qui est la dernière étape pour obtenir un modèle 3D.

Le terme **lumière codée** tire son sens de la codification des rayons lumineux sortant du projecteur. C'est cet encodage qui permet de retrouver facilement une correspondance dans l'image. Les intensités successivement récupérées dans un pixel de caméra *décrivent* la coordonnée pixel correspondante du projecteur. Notons que

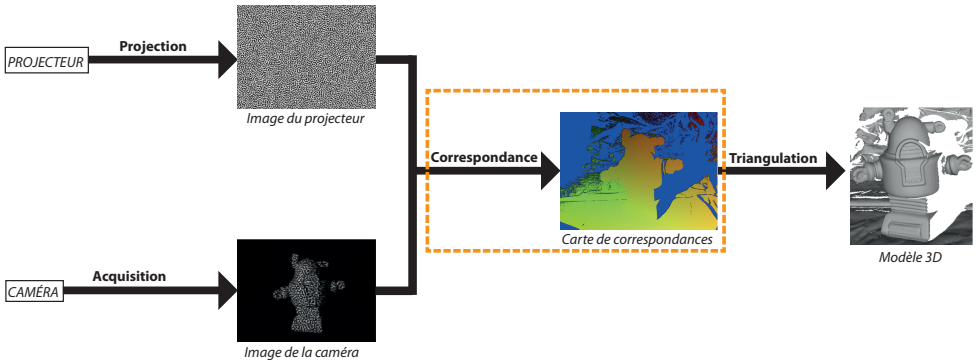


FIGURE 2.1: À partir de motifs projetés et capturés, les méthodes de reconstruction active calculent les correspondances entre le projecteur et la caméra afin de reconstruire un modèle 3D par triangulation. L'étape de mise en correspondance est cruciale, et c'est celle qui varie entre toutes les méthodes.

selon la méthode, la correspondance peut être déterminée directement ou indirectement, ce qui différenciera les méthodes dites de *lumière structurée* des autres.

La méthode la plus naïve de reconstruction active envoie autant d'images qu'il y a de pixels dans le projecteur. Pour chacune, un seul pixel serait éclairé et tous les autres éteints. Pour chaque pixel de la caméra, on pourrait directement associer son pixel correspondant dans le projecteur. Bien que prohibitive, cette méthode sera utilisée dans la section 8 pour produire des correspondances “optimales" à des fins de comparaison. Le but des méthodes de reconstruction active présentées dans ce chapitre est de récupérer les meilleures correspondances possible en projetant le moins d'images possible.

Dans cette section, nous présentons une revue des méthodes de lumière codée les plus utilisées. Dans un premier temps, nous survolerons les méthodes adaptées aux scènes dynamiques, requérant un minimum d'images à projeter. Nous nous attarderons davantage sur les méthodes temporelles par la suite. Ce sont ces méthodes que

nous utiliserons à des fins de comparaison tout au long de cette thèse.

Revue de littérature des méthodes de lumière codée

Les revues de littérature existantes sur les méthodes de reconstruction actives à l'aide d'un projecteur séparent souvent les méthodes en trois catégories : temporelles, spatiales ou directes[58, 57]. Ici, nous ne faisons la distinction qu'entre les méthodes requérant la projection d'une seule image et les méthodes *temporelles* qui nécessitent la projection d'une séquence de plusieurs motifs. Les méthodes de la première catégorie ont pour objectif la rapidité, souvent aux dépens de la qualité. Par symétrie, nous appelons ces méthodes **non temporelles**. La seconde catégorie de méthodes regroupe toutes celles nécessitant la projection de plus d'un motif. Ces méthodes dites **temporelles** utilisent l'information d'une séquence d'images pour déterminer les correspondances entre la caméra et le projecteur.

Bien que les méthodes non temporelles aient un intérêt particulier pour scanner des *scènes dynamiques*, i.e. dans lesquelles les objets peuvent être en mouvement, il serait incorrect de conclure que les méthodes temporelles ne peuvent pas fonctionner pour des objets en mouvement. La plupart des méthodes temporelles font la supposition que la séquence d'intensités récupérées pour un pixel de caméra correspond au même point dans la scène, ce qui peut être faux pour une scène dynamique. Cependant, si certaines suppositions sont faites, il est possible d'utiliser une méthode temporelle pour reconstruire des objets en mouvement[39, 49, 79]. Il est cependant clair que les méthodes non temporelles peuvent être utilisées pour des *scènes statiques*, bien qu'elles soient rarement utilisées dans ce contexte, puisque les méthodes temporelles produisent de bien meilleurs résultats dans ces conditions.

2.1 Méthodes non temporelles

Les méthodes de **codage spatial** utilisent le voisinage d'un pixel pour déterminer son code. En d'autres termes, l'intensité capturée pour un pixel n'est pas suffisante pour déterminer la correspondance dans le projecteur, il est nécessaire de connaître les intensités des pixels voisins. Ces méthodes requièrent souvent un processus compliqué pour la génération des motifs, puisque chaque voisinage de taille prédéfinie ne doit pas se répéter dans le projecteur pour garantir l'unicité des correspondances. Elles ont de la difficulté avec les scènes comportant beaucoup de discontinuités, puisque le voisinage peut lui-même être discontinu ou tout simplement en occlusion. Les reconstructions sont souvent clairsemées, c'est-à-dire que certains points ne sont pas reconstruits par manque d'information ou de confiance dans la correspondance trouvée. Cependant, ces méthodes n'utilisent souvent qu'une seule image pour obtenir une reconstruction et peuvent donc être utilisées pour des scènes dynamiques.

Une des premières méthodes[44] observe une image de segments verticaux de tailles aléatoires, et détermine le segment associé dans le projecteur en utilisant les tailles des 6 segments voisins. Cette méthode fonctionne très mal dans une scène contenant plusieurs discontinuités de profondeur. Une autre méthode utilise un motif composé de lignes horizontales formées par des séquences de 4 pixels dont les intensités sont choisies parmi 3 niveaux de gris[19]. Cet agencement particulier permet d'identifier les correspondances par autocorrélation. Cette méthode ne permet pas d'identifier uniquement chaque pixel, et suppose que chaque niveau de gris peut facilement être distinguable, ce qui n'est pas le cas sur des objets texturés. Dans [8], un motif basé sur une série de bandes verticales séparées par des bandes noires est proposée. Le motif est fait de sorte que si l'on retrouve dans l'image une séquence de pixels de couleurs entre deux bandes noires, il est possible d'identifier uniquement sa correspondance. Ici, chaque pixel peut être identifié uniquement, bien que l'algorithme de génération du motif soit fastidieux à mettre en place. De plus, la

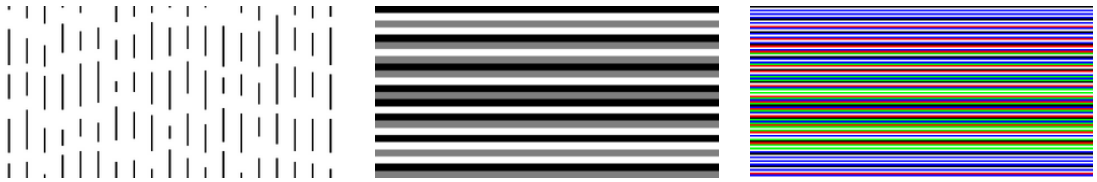


FIGURE 2.2: De gauche à droite, exemples de motifs utilisés par les méthodes de [44], [19] et [8]. Ces méthodes sont générées par des algorithmes ad hoc ou par essais et erreurs.

méthode peut se tromper lors de l'identification d'une séquence de couleurs, si certaines bandes ont disparu à cause d'occlusions, qui est un problème récurrent pour les méthodes spatiales. La figure 2.2 donne un exemple des motifs utilisés pour les méthodes spatiales mentionnées plus haut.

Une autre catégorie de codes spatiaux utilise les *séquences de De Bruijn*, pour résoudre le problème de l'unicité des codes. Une séquence de De Bruijn d'ordre m dans un alphabet de taille n est une chaîne circulaire de taille n^m qui contient toutes les séquences de taille m exactement une fois. Il existe des algorithmes déterministes pour générer ce type de séquence. Le décodage de ces motifs est plus robuste que pour les méthodes non formelles, car la garantie d'unicité du voisinage réduit les chances d'erreurs de correspondance et simplifie le processus de génération du motif. Une des premières méthodes utilisant ces séquences[69], construit un motif binaire basé sur l'entrelacement de deux séquences de De Bruijn. Dans ce motif, chaque paire de lignes horizontales est formée à partir de bits extraits de l'entrelacement des deux séquences, garantissant l'unicité de chaque fenêtre de 2×3 pixels le long d'une ligne, mais pas entre les lignes. La plupart des méthodes utilisent plutôt des images en couleur, qui permettent d'agrandir la taille de l'alphabet et donc de garantir une unicité dans un plus grand voisinage. Une de ces méthodes utilise un motif composé de 125 bandes de 8 couleurs différentes déterminées par les symboles d'une séquence de De Bruijn[77]. Une des contributions majeures de ces travaux pour les méthodes

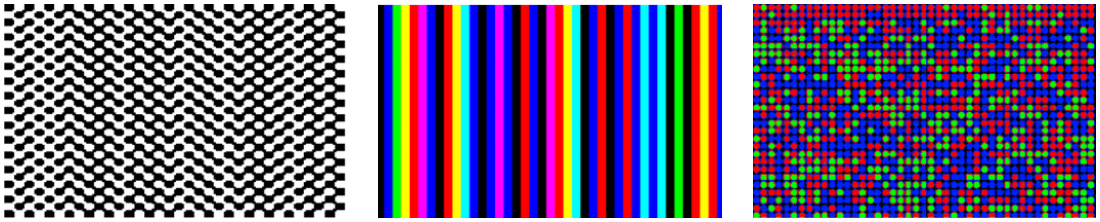


FIGURE 2.3: À gauche et au centre, exemples de motifs utilisés par les méthodes de [69], [77] basées sur des séquences de De Bruijn. À droite, motif utilisé par la méthode de [47] basée sur les tableaux M.

spatiales est un algorithme de programmation dynamique qui identifie chaque bande selon son voisinage, en prenant en compte le fait que des bandes peuvent disparaître ou ne pas être dans le même ordre que dans l'image du projecteur à cause des occlusions ou des discontinuités de profondeur. Les correspondances ne peuvent pas toujours être récupérées de manière dense, dû à la largeur des bandes qui n'identifient pas uniquement chaque pixel. L'auteur présente aussi une extension temporelle qui permet d'améliorer la résolution en déplaçant latéralement le motif d'un pixel sur une distance qui couvre la plus grande largeur de bande. Il s'agit d'un procédé souvent adopté dans les méthodes temporelles pour améliorer la précision des reconstructions. Les séquences de De Bruijn sont aussi utilisées pour créer des grilles 2D dont les lignes et colonnes utilisent un ensemble de couleurs différentes, garantissant une unicité en 2D[56, 36]. Bien que garantissant l'unicité des codes dans un voisinage, ces méthodes produisent rarement des reconstructions denses (à l'exception de l'extension de [77]) et nécessitent des méthodes complexes pour résister aux discontinuités de profondeur et aux occlusions. Un exemple des motifs utilisés pour les méthodes basées sur des séquences de De Bruijn est présenté à la figure 2.3.

Les *tableaux M* sont des matrices pseudo-aléatoires de symboles dans un alphabet de k éléments possédant la *propriété de fenêtre* $n \times m$, c'est-à-dire que chaque sous-matrice de taille $n \times m$ apparaît une seule fois dans la matrice (à l'exception de

la sous-matrice de 0). C'est une autre méthode utilisée pour générer des motifs dont le voisinage est unique. Les motifs basés sur cette méthode de génération peuvent encoder les éléments de l'alphabet avec des symboles[24] ou des cercles de couleurs [47] (comme à la figure 2.3). Il est possible d'inclure de la redondance dans la représentation et intégrer des codes correcteurs directement dans chaque sous-fenêtre pour réduire les erreurs de décodage. La mise en correspondance passe par une étape de segmentation pour isoler chaque sous-fenêtre, puis une étape de décodage pour identifier correctement chacune d'entre elles. Ces méthodes ne peuvent également pas générer des reconstructions denses, puisque chaque sous-fenêtre de pixels est unique, mais chaque pixel de la fenêtre est indissociable de ses voisins. Elles ne produisent pas de très bons résultats pour des scènes fortement discontinues, et sont sensibles aux scènes texturées et colorées.

2.2 Méthodes temporelles

Selon notre distinction, cette catégorie de méthodes regroupe les méthodes à multiplexage temporel, et à codage direct de la classification de [58]. Les méthodes à **codage direct** utilisent normalement moins d'images que les méthodes strictement temporelles. En effet, elles utilisent souvent un ratio d'intensité entre deux ou trois images d'un pixel éclairé pour déterminer sa correspondance. Par conséquent, ces méthodes sont beaucoup plus sensibles au bruit et à la texture présente dans la scène que les méthodes qui utilisent plusieurs motifs de projection.

La première méthode, introduite par [10], utilise une rampe de niveaux de gris de blanc à noir. La correspondance d'un pixel est déterminée par le ratio entre l'intensité mesurée avec et sans illumination par ce motif. Cette méthode ne fonctionne pas particulièrement bien, car le ratio est très bruité et dépend d'un calibrage photométrique parfait entre les couleurs projetées par le projecteur et celles mesurées par la caméra. Cette méthode a été reprise à plusieurs reprises afin d'en améliorer

les performances[46, 12]. De fait, la méthode pyramidale de [12] partage énormément de ressemblances avec les codes binaires présentés plus tard. Elle nécessite la projection d'une quantité plus grande de motifs, sans pour autant produire de meilleurs résultats.

Il existe aussi des méthodes de codage direct qui utilisent la couleur plutôt que des niveaux de gris. La méthode de [59] utilise trois images d'un motif d'arc-en-ciel périodique, chacune étant déplacée d'un tiers de sa période. Comme la plupart des méthodes utilisant des motifs périodiques, lors de l'identification du pixel correspondant, il existe une ambigüité qu'il faut résoudre. Ce phénomène est expliqué plus en détail dans la section concernant les méthodes à déphasage.

Les méthodes basées sur un **codage temporel** utilisent la succession temporelle des motifs projetés pour encoder la position du projecteur. Par conséquent, elle nécessite souvent la projection d'une quantité non négligeable de motifs pour fonctionner. Par contre, elles se distinguent par la précision et la robustesse des reconstructions qu'elles produisent. De plus, puisqu'elles n'utilisent pas le voisinage d'un pixel pour en déterminer sa correspondance, elles produisent une reconstruction dense, pour laquelle chaque pixel de caméra peut être reconstruit en 3D indépendamment de ses voisins.

Les approches basées sur un *codage binaire* ont été les premières méthodes temporelles explorées. Dans la version originale [53], chaque pixel du projecteur est encodé par sa notation binaire, c'est-à-dire que la i^{e} image est composée à partir des valeurs du i^{e} bit de la notation binaire de chaque pixel. Il suffit alors de projeter n images pour obtenir 2^n codes différents. En pratique, il faut n images par axe afin d'obtenir une coordonnée pixel en x et en y. Lors du décodage, décider si un pixel de l'image a été illuminé ou non est le principal problème de ces méthodes. Projeter une image et son inverse est normalement la solution : on décide qu'un pixel est allumé s'il est plus foncé que son inverse. Cela ne fonctionne cependant pas dans les zones d'occlusion ni sur les surfaces très réfléchissantes et cela double le nombre d'images. Ainsi, chaque

fois qu'un pixel est déclaré illuminé alors qu'il ne l'est pas, cela produit une erreur d'un bit, qui peut complètement changer la correspondance décodée. Une variante basée sur les *codes de Gray* introduits par [33], qui ont l'avantage de différer exactement de 1 bit entre chaque pixel voisin, ajoute une certaine robustesse aux erreurs de décodage (voir figure 2.4).

Les motifs correspondants aux bits de poids faible sont souvent très difficiles à retrouver, car ils ont tendance à fusionner dans l'image de la caméra et être capturés comme une image de gris moyen. La méthode de [68] propose de détecter les bandes avec une précision sous-pixel, en cherchant la position du passage par zéro dans le laplacien de l'image de la caméra. Une autre solution pour récupérer les derniers bits avec une bonne précision et de projeter un motif composé de lignes parallèles que l'on déplace sur plusieurs pixels afin de couvrir toutes les positions dans l'image[26, 67] (voir figure 2.4). La méthode proposée par [26] combine les codes de Gray ainsi qu'un motif contenant une ligne blanche tous les six pixels. L'intersection d'une ligne horizontale et verticale donne directement la position sous-pixel d'un pixel de projecteur dans l'image de la caméra. Les codes de Gray sont utilisés pour lever l'ambigüité quant à la position absolue du pixel de projecteur. En effet, chaque intersection indique seulement la paire d'images parmi les 6×6 possibles, et il faut ensuite décider de quelle ligne horizontale et verticale dans la paire il s'agit. La méthode de [67] quant à elle, utilise le défocus naturel introduit par la projection d'une ligne à l'aide d'un projecteur pour estimer la position sous-pixel dans le projecteur de chaque pixel de caméra. La distance sous pixel à la ligne de projection est estimée en ajustant une gaussienne sur les intensités récupérées pour chaque motif de ligne déplacée. Ces deux méthodes sont donc complémentaires, bien que l'une reconstruise du point de vue du projecteur, l'autre du point de vue de la caméra.

Il est bien sûr possible d'utiliser plus que deux illuminations pour encoder les codes binaires. Par exemple, plusieurs tons de gris ont été utilisés ainsi que des motifs en couleur[11]. L'avantage majeur est que le nombre d'images à projeter diminue, en

dépit d'une plus grande sensibilité au bruit et à la nécessité de réaliser un calibrage photométrique précis. Le modèle d'illumination proposé par [11] pour obtenir la relation entre les couleurs projetées et capturées, peut être utilisé pour toute méthode qui utilise des motifs en couleur.

Une autre approche démocratisée par [75] utilise le déphasage de motifs sinusoïdaux périodiques pour encoder la position d'un pixel de projecteur. Une fonction sinusoïdale est déphasée plusieurs fois, puis discrétisée et finalement projetée (voir figure 2.4). La caméra observe pour chaque pixel plusieurs intensités lors de chaque déphasage, et ces intensités sont reliées à la phase du pixel de projecteur correspondant. Le strict minimum est d'utiliser trois déphasages afin de pouvoir calculer la phase d'un pixel[75, 79]. Ainsi, il est possible de ne projeter qu'une seule image en couleur dont chaque canal est un des sinus déphasés. Avec une seule image, cette méthode est très sensible à plusieurs facteurs comme la réflectance de la surface et les propriétés du projecteur. Un gros désavantage des méthodes à déphasage est l'ambigüité existante entre deux pixels de projecteur ayant la même phase, mais dans deux périodes différentes. Sans informations additionnelles, il est impossible de déterminer la période, et donc chaque phase est calculée à un modulo près. La méthode à trois déphasages par exemple, ne peut fonctionner pour des scènes discontinues ou contenant des objets dont les différences de profondeur sont trop grandes. C'est un problème cependant récurrent pour toutes les méthodes qui utilisent des motifs périodiques, peu importe le nombre de déphasages utilisés. Ce problème est souvent résolu par un algorithme de *désambigüisation de la phase*. Plusieurs algorithmes ont été proposés pour résoudre ce problème[81, 34, 32]. L'algorithme présenté dans [81] projette plusieurs sinus de fréquences différentes, chacun déphasé plusieurs fois. La phase calculée pour chaque fréquence est ensuite utilisée comme "point de départ" pour déterminer la phase dans le sinus à plus haute fréquence. On utilise ainsi chaque fréquence jusqu'à la plus haute, qui permet de retrouver une phase absolue (i.e. qui n'est plus ambigüe) et très précise.

Dernièrement, plusieurs méthodes ont été proposées pour résister aux problèmes de l'illumination indirecte[27, 13, 28]. Les phénomènes causés par l'illumination indirecte sont expliqués en détail au chapitre 3. Dans ce chapitre, nous indiquons que les méthodes utilisant des motifs de basse fréquence, tels que les bits de poids fort des codes de Gray (voir figure 2.4) sont très sensibles à l'illumination indirecte. La méthode proposée par [27] s'inspire de ce fait pour modifier les codes de Gray en n'utilisant que des motifs hautes fréquences. Pour ce faire, chaque motif des codes de Gray est "modulé" par un motif haute fréquence dans l'image du projecteur et de la caméra. Ainsi, les bits de poids fort ne sont jamais projetés, mais ils peuvent être retrouvés en démodulant les motifs projetés par le même motif haute fréquence. En particulier, les auteurs utilisent la fonction `XOR` comme opérateur de modulation, et le motif haute fréquence choisi peut être le motif correspondant aux bits de poids le plus faible, par exemple. Plusieurs ensembles de motifs sont projetés, utilisant diverses hautes fréquences et la correspondance d'un pixel est choisie à majorité entre les résultats de chaque ensemble de motifs (voir figure 2.4). Le principe de la modulation avait déjà été utilisé dans la méthode à déphasage modulé[13]. Pour obtenir des correspondances sur des surfaces très réfléchissantes, les auteurs ont proposé de moduler les motifs de la méthode à déphasage de signaux sinusoïdaux par un motif haute fréquence. Ainsi chaque motif projetté est modulé par un sinus haute fréquence dans la direction opposée (voir figure 2.4). Pour retrouver la phase, il faut en premier lieu démoduler l'image capturée par la caméra, puis appliquer la méthode originale de résolution de la phase sur les motifs démodulés. Le principal désavantage de cette méthode est le nombre d'images requis pour effectuer la modulation. En effet, à la différence de [27] où chaque motif est modulé une seule fois par un code binaire haute fréquence, chaque motif de la méthode de [13] nécessite plusieurs déphasages pour être modulé correctement. Une dernière méthode produisant de très bons résultats est la méthode de micro déphasage[28]. Cette méthode se base sur le même principe d'utilisation de signaux sinusoïdaux hautes fréquences uniquement. L'avantage

principal de cette méthode est la possibilité de résoudre le problème de désambigüisation de la phase à l'aide de signaux uniquement hautes fréquences. La méthode que nous présentons au chapitre 4 est la seule avec la méthode de micro déphasage à obtenir des correspondances sans utiliser de basses fréquences, garantissant une très grande robustesse à l'illumination indirecte. Un autre avantage de la méthode est le nombre minimal d'images requises pour obtenir une correspondance (7 images sont suffisantes). Cependant, un filtre médian est nécessaire pour résister au bruit. Nous montrons dans le chapitre 6 que ce filtre peut éliminer des correspondances pertinentes pour les petits objets. Un autre désavantage de cette méthode par rapport à la nôtre, est que les signaux sinusoïdaux sont hautes fréquences uniquement dans une direction. La direction opposée est basse fréquence et donc sensible aux interréflexions, bien que dans une proportion moindre que les images basses fréquences dans les deux dimensions.

La figure 2.4 regroupe des exemples de motifs utilisés par les méthodes temporelles décrites dans ce chapitre. Elle montre aussi les motifs des méthodes que nous présentons aux chapitres 4 et 8. Toutes les méthodes qui utilisent des motifs contenant de larges bandes blanches auront des difficultés dues à l'illumination indirecte. Les méthodes de la dernière ligne de la figure 2.4 sont les seules qui possèdent des fréquences hautes dans les deux directions, ce qui explique leur robustesse accrue par rapport aux autres méthodes. Dans le chapitre 3, nous introduirons les défis des méthodes de lumière codées et au chapitre 4, nous présenterons la méthode basée sur les motifs de lumière non structurée.

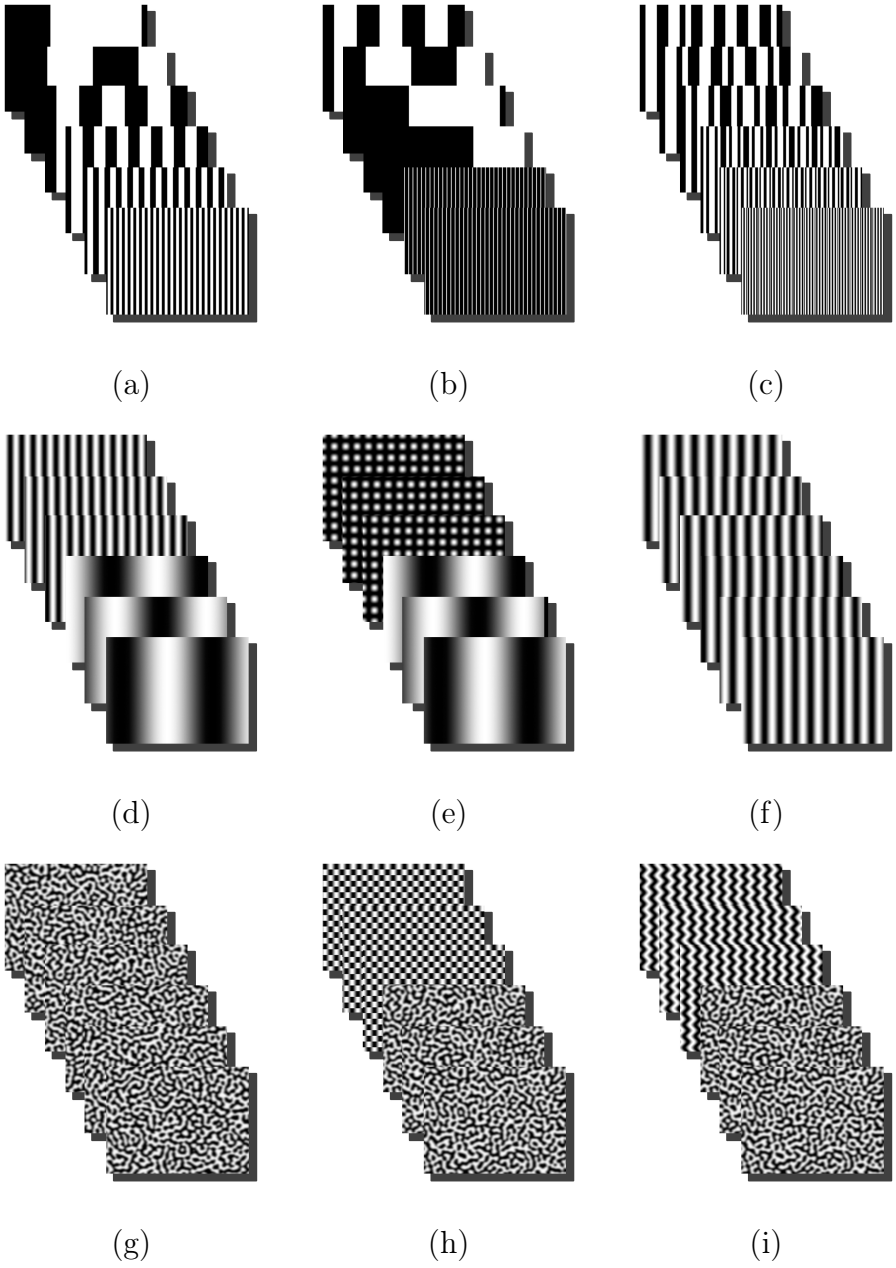


FIGURE 2.4: Exemples de motifs utilisés par des méthodes temporelles. (a) codes de Gray[33], (b) déphasage de lignes[67], (c) ensemble de codes `XOR`[27], (d) déphasage sinusoïdal[81], (e) déphasage sinusoïdal modulé[13], (f) micro déphasage[28], (g) lumière non structurée[42], et méthodes hybrides (h) et (i). Les méthodes (g), (h) et (i) sont présentées respectivement aux chapitres 4 et 8.

Première partie

Robustesse des méthodes de reconstruction active

Chapitre 3

LES PRINCIPAUX DÉFIS DE LA LUMIÈRE CODÉE

La mise en correspondance des images capturées par une caméra d'une scène illuminée par un projecteur fait face à plusieurs problèmes. En effet, les défis posés par les méthodes de lumière codée peuvent être dus aux propriétés photométriques du projecteur et de la caméra, ou bien à celles de la scène.

3.1 Propriétés photométriques de la caméra et du projecteur

Les motifs projetés dans une scène sont observés par la caméra sous la forme d'intensités mesurées. Le lien entre ces intensités et le motif dépend de plusieurs paramètres. Pour simplifier la présentation, supposons que la scène ne contient que des objets lambertiens (c'est-à-dire des objets dont l'intensité perçue ne dépend pas de la position de l'observateur, cf. section 1.1). Supposons que les intensités soient comprises entre les valeurs 0 et 1. Dans ce cas, la relation entre l'intensité I_p émise par un projecteur, et l'intensité I_c capturée par une caméra est donnée par[80, 50] :

$$I_c = (\alpha \, ({I_p}^{\gamma_p} + I_g) + \beta)^{\gamma_c} \tag{3.1}$$

où α représente l'*albédo* de la surface au point d'impact du rayon lumineux sur l'objet, β représente l'éclairage ambiant et γ_p et γ_c modélisent respectivement les non-linéarités du projecteur et de la caméra dues à la *correction gamma*[74]. Dans le modèle lambertien, l'albédo est le coefficient de réflectance (quantité de lumière réfléchie par l'objet) en un point de la surface[41], qui varie pour chaque point de la scène. La majorité des méthodes de reconstruction active fonctionnent moins bien pour les objets dont l'albédo est faible, car les intensités mesurées pour ces points sont petites et donc plus sensibles au bruit. L'albédo peut être estimé facilement[50]

bien que ce soit rarement nécessaire. L'éclairage ambiant est composé aussi bien de l'éclairage global présent lorsque le projecteur est éteint, que de l'éclairage résiduel lorsque le projecteur projette une image noire du fait de son contraste limité. Le terme I_g représente l'apport de l'illumination globale à l'intensité perçue par une caméra pour le point éclairé directement par le projecteur. Il s'agit de la somme de toutes les illuminations indirectes (e.g. après rebond sur une surface réfléchissante, passage à travers une surface transparente, etc., voir la figure 3.1). Finalement, les non-linéarités γ_c et γ_p résultent de la conversion des signaux électriques lumineux en valeurs numériques. En effet, l'oeil humain n'a pas la même sensibilité aux intensités claires qu'aux intensités foncées. De fait, un projecteur ne projette jamais les intensités d'un motif de manière linéaire, mais compense plutôt cette sensibilité en appliquant une fonction γ [74]. De même, la plupart des caméras transmettent leurs images après passage dans un espace de couleur adapté à la numérisation des niveaux de gris, et adaptent les intensités pour minimiser la perte lors de la conversion. Il existe des méthodes pour estimer ce paramètre aussi bien dans le projecteur que dans la caméra[40].

Certaines méthodes sont invariantes à ces paramètres. En particulier, la plupart des méthodes qui n'utilisent que deux intensités 0 et 1, puisqu'elles ne sont pas affectées par les non-linéarités. C'est le cas de la méthode que nous présentons dans le prochain chapitre. Puisqu'elle utilise la différence d'intensités entre deux motifs capturés pour un même pixel, en plus de la robustesse au gamma, elle n'est sensible à aucun des paramètres de l'équation 3.1. En effet, bien que non linéaire, cette équation est monotone, et c'est une propriété suffisante pour que notre méthode fonctionne sans calibrage photométrique préalable (i.e. elle garantit que le signe de la différence entre deux intensités est le même dans la caméra et le projecteur). Les méthodes qui utilisent des niveaux de gris et supposent la linéarité du processus d'acquisition des intensités doivent corriger les motifs avant la projection et après capture, pour en tenir compte. Lorsque les méthodes utilisent des motifs en couleur, il faut en plus

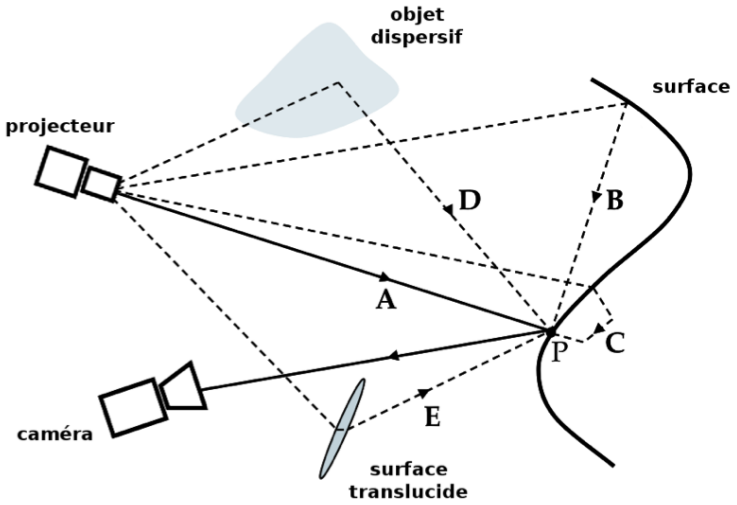


FIGURE 3.1: La radiance d'un point P de la scène dépend de l'illumination directe du projecteur (A) et de l'illumination indirecte qui inclut l'interréflexion (B), la dispersion sous-surface (C), la dispersion volumétrique (D) et la translucidité (E). Tirée de [14] (traduction libre de [50]).

modéliser et calibrer adéquatement le transfert des couleurs pour chaque canal rouge, vert et bleu entre le projecteur et la caméra[11].

3.2 Propriétés photométriques de la scène

La configuration spatiale de la scène ainsi que les matériaux qui la composent ont une grande influence sur le bon fonctionnement d'une méthode de reconstruction active. En effet, lorsqu'un point de la scène est éclairé par le projecteur, la radiance en ce point n'est pas seulement composée de l'éclairage direct, mais aussi des sommes des éclairages indirects provenant des autres points de la scène (voir la figure 3.1). Les travaux de [50] ont mis en évidence la nécessité d'adapter les motifs à projeter en fonction des matériaux des objets de la scène. Un exemple de scène composée de matériaux difficiles à reconstruire est présenté au chapitre 6 à la figure 6.9. Les

matériaux comme la cire ou la peau humaine absorbent et dispersent la lumière à l'intérieur de l'objet et sont particulièrement difficiles à reconstruire. Les interréflexions sont causées par le rebond de la lumière sur des surfaces particulièrement réfléchissantes ou du fait de la proximité des objets de la scène (e.g. un coin de mur).

L'illumination indirecte produite par tous ces phénomènes pose de très gros problèmes aux méthodes de lumière codée standards (voir figure 4.2 au chapitre 4). La difficulté vient du fait que pour fonctionner correctement, seulement l'illumination directe devrait être prise en compte. Plusieurs méthodes ont été proposées pour séparer systématiquement les composantes directes et indirectes[62, 5, 50] de l'illumination. En pratique, il est peu commode d'appliquer ces méthodes pour ne conserver que la composante directe des pixels pour *chaque* motif projeté. Il a cependant été suggéré que les motifs composés uniquement de hautes fréquences étaient adaptés pour extraire la composante indirecte[50]. Lorsqu'un motif est composé de fréquences suffisamment hautes, chaque zone de la scène illuminée reçoit une quantité similaire de blanc et de noir. Par conséquent, une méthode temporelle peut considérer qu'une séquence de motifs hautes fréquences produit toujours la même quantité d'illumination indirecte pour un point de la scène, ce qui résulte en un terme constant dans l'équation 3.1. Inversement, lorsque de larges zones blanches sont projetées, les interréflexions de la scène ne peuvent plus être ignorées (voir figure 6.6).

Dans le prochain chapitre, les motifs que nous utilisons sont *bandes passantes*, c'est-à-dire, composés uniquement de fréquences dans un intervalle adapté à la scène afin d'homogénéiser l'illumination indirecte pour pouvoir la considérer constante. Ces motifs, dit *non structurés*, ne dépendent pas d'un calibrage photométrique ni géométrique, et utilisent des fréquences assez élevées pour résister aux défis posés par l'illumination indirecte présente dans la scène.

Chapitre 4

UNSTRUCTURED LIGHT SCANNING ROBUST TO INDIRECT ILLUMINATION AND DEPTH DISCONTINUITIES (ARTICLE)

Ce chapitre présente l'article[16] publié tel que l'indique la référence bibliographique :

> V. Couture, N. Martin et S. Roy, Unstructured light scanning robust to indirect illumination and depth discontinuities, International Journal of Computer Vision, (2014), p. 1–18.

Il fait suite à une publication dans la conférence *IEEE Computer Society International Conference on Computer Vision (ICCV) 2011* [15].

Cet article présente une méthode de reconstruction active utilisant des patrons de lumière non structurée. Bien que la lumière non structurée ait déjà été utilisée [35], elle est utilisée ici dans un contexte de réduction des interréflexions.

Les motifs sont générés à partir d'une image de bruit blanc dont les fréquences ont été filtrées pour ne garder qu'un certain intervalle, se soldant par une image qui ne contient ni basses fréquences, responsables de la majorité de l'illumination indirecte[50], ni très hautes fréquences. À la différence des patrons de lumière structurée, la position d'un pixel de projecteur n'est pas encodée à travers la séquence de motifs projetés, et une méthode de mise en correspondance spéciale est présentée afin d'établir l'appariement entre pixels de caméra et projecteur efficacement.

La méthode est comparée aux méthodes de lumière structurée standard [33, 79] ainsi qu'à une méthode robuste aux interréflexions [27].

L'article est présenté dans sa version originale.

Abstract

Reconstruction from structured light can be greatly affected by indirect illumination such as interreflections between surfaces in the scene and sub-surface scattering. This paper introduces band-pass white noise patterns designed specifically to reduce the effects of indirect illumination, and still be robust to standard challenges in scanning systems such as scene depth discontinuities, defocus and low camera-projector pixel ratio. While this approach uses *unstructured* light patterns that increase the number of required projected images, it is up to our knowledge the first method that is able to recover scene disparities in the presence of both indirect illumination and scene discontinuities. Furthermore, the method does not require calibration (geometric nor photometric) or post-processing such as phase unwrapping or interpolation from sparse correspondences. We show results for a few challenging scenes and compare them to correspondences obtained with the well-known Gray code and Phase-shift methods, and with the recently introduced method by Gupta *et al.*, designed specifically to handle indirect illumination.

4.1 Introduction

Scene reconstruction from structured light is the process of projecting a known pattern onto a scene, and use a camera to observe the deformation of the pattern to calculate surface information. The term "structure" comes from the fact that a unique code (a finite set of patterns) is associated to each projector pixel, based on its position in the pattern. Camera-projector pixel correspondence (see Fig. 4.1) can then directly be established and triangulated to estimate scene depths. Results produced by structured light scanning systems greatly depend on the scene and the patterns used. In particular, it was shown in [50] that low frequency patterns create interreflections in scene concavities that cannot be removed. Another issue comes from scene depth discontinuities, where smoothness of the observed pattern can no

longer be assumed.

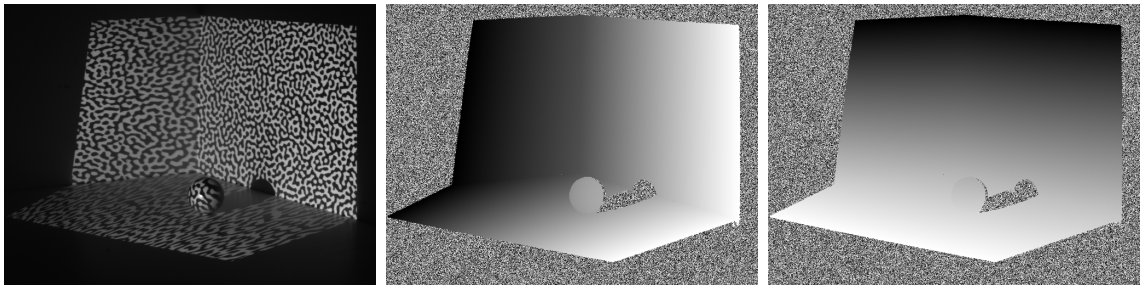


FIGURE 4.1: Example of a scene (left) with one unstructured band-pass pattern projected on it. Several of these patterns are used to recover the x (center) and y (right) correspondence maps between the camera and the projector.

In this paper, we propose the use of band-pass white noise patterns that are specifically designed to reduce the effects of indirect illumination [1] while still being able to handle depth discontinuities. These patterns follow the basic idea of *unstructured* light patterns [35, 17, 71] that do not directly encode pixel position in the projector. Their only restriction is that the accumulation of such patterns uniquely identifies every projector pixel. Therefore, the correspondence of a camera pixel is no longer computed directly from the observed pattern sequence, and has to be found using an iterative high-dimensional matching algorithm. The matching method we present here is not limited to epipolar lines to avoid the need to geometrically calibrate any of the devices in order to recover correspondence.

The spatial frequency of these patterns can be adjusted, making them robust to defocus (due to small depth of field, for instance) or low camera-projector pixel ratio [2]. Also, the method is designed to be independent of photometric properties (such as gamma correction) of both the projector and the camera.

1. In the literature, indirect illumination is sometimes called *global* illumination.

2. The camera-projector pixel ratio is defined as one camera pixel over the number of projector pixels it can see.

The method was first presented in [15] specifically to address the problem of interreflections. Here, we include new results to show that the method also works for other types of indirect illumination such as translucency and sub-surface scattering. We also compare our results with those of other methods, namely the Gray code and Phase-shift methods, and a recently introduced method by Gupta *et al.* [27, 28] to handle indirect illumination.

The layout of this paper is as follows. We begin in Sec. 4.2 by briefly reviewing prior works related to structured light patterns. We then expose in Sec. 4.3 common problems that may arise in structured light setups, namely indirect illumination, scene depth discontinuities and a low camera-projector pixel ratio. In Sec. 4.4, we introduce unstructured band-pass white noise patterns and discuss their properties. Using these patterns, matching between projector and camera pixels requires a high-dimensional match algorithm, namely locally sensitive hashing, which we describe in Sec. 4.5. In Sec. 4.6, the Gupta *et al.* method that also handles indirect illumination is reviewed. Finally, we compute in Sec. 4.7 camera-projector correspondence maps and reconstructions using our unstructured light patterns and compare results produced by other methods for different challenging scenes. We conclude in Sec. 4.8.

4.2 Previous work

Several sets of structured light patterns were previously proposed to perform active 3D surface reconstruction. Structured light reconstruction are often classified based on the type of encoding used in the patterns : temporal, spatial or direct [58]. Here, we also emphasize the amount of supplemental information needed by the method to work effectively. For instance, prior photometric or geometric calibration is often required.

Temporal methods multiplex codes into pattern sequence[53, 60, 33, 26]. For instance, a pixel position is encoded in [45, 53] by its binary code, representated by a

concatenation of binary coded patterns. One variation introduces Gray code patterns [33] that are designed to minimize the effect of bit errors by ensuring that neighboring pixels have a code difference of only one bit. Temporal methods require a high number of patterns and the scene must remain static during the pattern acquisition process. In practice, these methods can give very good results and do not require any kind of calibration. Due to focus issues or low pixel ratio, the lowest significant bits often cannot be recovered. Solutions have been proposed, like in [26] where high frequency patterns are replaced by a shifted version of a pattern to recover the last significant bits. This method (and all variants of binary encoding patterns) also suffers from the significant indirect lighting induced by the lower frequency patterns, as we will see in the next section.

In contrast, spatial methods use the neighborhood of a pixel to recover its code [8, 69, 56] in order to decrease the number of required patterns. For example, the patterns can be stripes [8], grids [54] or a more complicated encoding such as the popular De Bruijn patterns [69]. Except for grids, it is worth mentioning that these patterns are one-dimensional, and thus require a geometric calibration relating the camera and the projector. Some methods even allow "one-shot" calibration [56] (i.e. only one pattern is used), but they require a very good photometric calibration. The main drawback of these methods is that they assume spatial continuity of the scene, which does not hold at depth discontinuities. Furthermore, those methods produce sparse results, as the correspondence can be recovered only at stripe transitions of the pattern. In [77], high quality reconstructions of static scenes are computed using a multi-pass dynamic programming edge matching algorithm. The pattern is shifted over time to compensate for the sparseness of De Bruijn patterns. The number of patterns required is still a lot less than in the case of temporal methods. However, the method requires both photometric and geometric calibration.

Direct coding methods use the intensity measured by the camera to directly estimate the corresponding projector pixel. Similarly to temporal methods, no spatial

neighborhood is required to obtain correspondence. Direct methods need only a few patterns, typically three patterns. Because patterns can be embedded in a single color image, one image is theoretically sufficient to recover depth. The work of [75] introduced the so-called "three phase-shift" method which relies on the projection of three dephased sinusoidal patterns. This method was modified in [79] to project only two sinusoidal patterns and a neutral image used as a texture. These methods often require the estimation of the gamma coefficient (for both the projector and the camera) and, because they are one-dimensional, a geometric calibration as well. More patterns can also be used to modulate the signal in 2D and reduce the effects of noise and gamma factors [13]. Furthermore, matching using these patterns is ambiguous due to their periodic nature. In practice, phase unwrapping is used to overcome this issue, but high frequency patterns remain ambiguous for scenes with large depth discontinuities.

We present in Sec. 4.4 a novel temporal method that uses *unstructured* light patterns that are not dependent on projector pixel position. Similar work has been presented in [35] where scanning is performed using a sequence of photographs or a sequence of random noise patterns for flexibility purposes. Contrary to [35] however, we designed the unstructured patterns specifically to minimize the effects of indirect illumination. Another method was recently introduced in [27] to address the problem of indirect illumination using a combination of high frequency patterns, band-pass patterns and standard Gray codes. We will compare this method with our approach in Sec. 4.6. Our method will also address typical challenges that may arise in structured light setups. We review these in the following section.

4.3 Problems of structured light systems

This section reviews the problems that may arise in typical structured light setups, such as indirect lighting, varying camera-projector pixel ratios, and scene depth

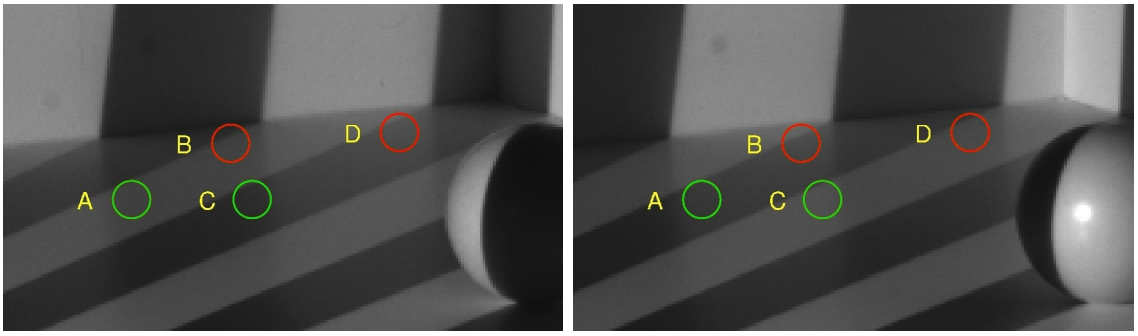


FIGURE 4.2: A stripe pattern (left) and its inverse (right) are displayed. Measured intensities at points (A, B, C, D) are $(56, 56, 35, 71)$ and $(46, 66, 72, 65)$ in the left and right images respectively. Points B and D are incorrectly classified because of interreflection.

discontinuities. It also discusses strengths and weaknesses of the methods reviewed in Sec. 4.2.

4.3.1 Indirect illumination

When a scene is lit, the radiance measured by the camera has two components, namely direct illumination due to direct lighting from the projector and indirect illumination caused by light reflected from or scattered by other points in the scene for instance[50]. It is generally assumed that when projecting a Gray (or binary) code pattern followed by its inverse, a camera pixel is lighter when observing a white stripe [58]. This is not always the case however, especially in the presence of indirect illumination, as illustrated in Figure 4.2 by points B and D. This situation severely deteriorates the quality of the recovered codes.

Nayar *et al.* presented in [50] a method to separate direct and indirect components of illumination. They showed that indirect illumination becomes a constant gray intensity when the pattern frequency is high enough, i.e. that geometry, reflectance map and direct illumination are smooth with respect to the frequency of the

illumination pattern. Separation is done by subtracting the image of a single high frequency binary pattern and its complement, or by subtracting the minimum from the maximum intensities measured over a few patterns.

Structured light methods that use only high frequency patterns could potentially remove the effects of indirect lighting to improve performance. Phase-shift methods are good examples, but increasing the frequency also increases signal periodicity, which makes the subsequent phase unwrapping step hard if not impossible to accomplish. Therefore, lower frequency patterns tend to be used in practice [58].

For low frequency patterns, it is much harder to remove the effects of indirect illumination. A few methods were proposed to partially achieve this by modulating low frequency patterns with high frequency patterns [13, 25, 27]. Indirect lighting could also be estimated using a light transport matrix [52, 37, 23] which relates every pixel of the projector to every pixel of the camera. However, this matrix is huge and very time consuming to measure and process. For illustration purposes, we computed this matrix, which was then transposed and remapped from projector to camera using our matching results. Figure 4.3 shows how different regions in the scene contribute to the intensity measured at selected camera pixels by creating indirect lighting. As in [50], we argue that if the pattern spatial frequency is high enough, then these contributing areas always include an equal mixture of black and white, thereby making indirect lighting near constant.

4.3.2 Depth discontinuities

Spatial methods such as De Bruijn patterns require a neighborhood around a pixel to estimate its code. This allows a reduction in the number of patterns, but creates problems near depth discontinuities where the camera observes a mixture of at least two projector pattern regions. This makes decoding unstable. For this reason, spatial methods require a post-processing step to remove wrong matches near discontinuities, usually a dynamic-programming minimization to add smoothness constraints on the

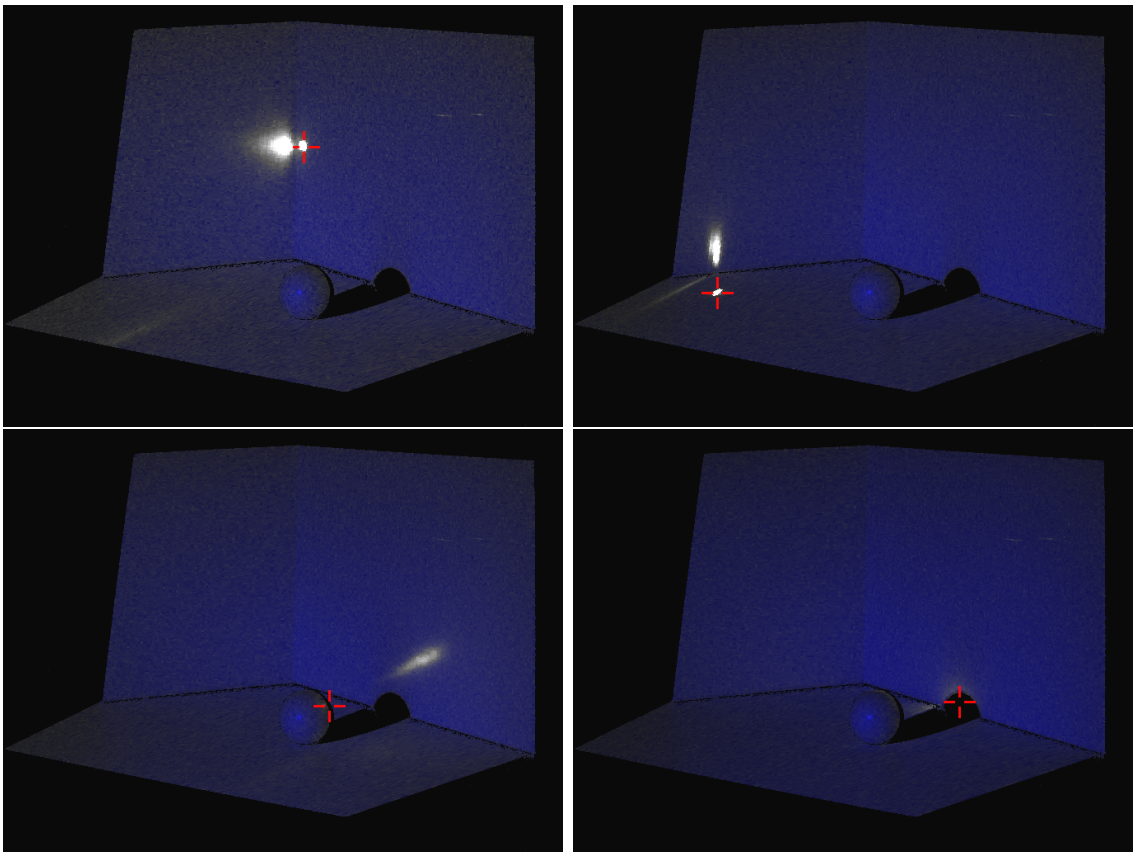


FIGURE 4.3: Illumination contribution for selected pixels, indicated by red crosshairs. The blue color is added artificially to provide a scene reference. Top has direct lighting with interreflections. Bottom left feature indirect lighting. Bottom right is a pure shadow.

correspondence map [77].

For temporal and direct methods, which do not require any spatial neighborhood, correspondence errors can occur when two codes at different depths are both seen by the same camera pixel. This blends two unrelated codes and affects direct methods such as Phase-shift which rely on the measured intensity to estimate correspondence.

4.3.3 Pixel Ratio

Because of the relative geometry and resolution of the camera and projector, it is often the case that a single camera pixel captures a linear combination of two or more adjacent projector pixels. This situation often occurs in multi-projector setups, where the total resolution of the projectors is far greater than the camera resolution. This is known as having a low camera-projector pixel ratio.

The Gray code method degrades gracefully with pixel ratio, as low significant bits become too blurred to be recovered and are simply discarded. Other methods, such as De Bruijn or Phase-shift, are robust to this as long as their pattern frequencies are low enough.

4.4 Unstructured light patterns

This section presents our *unstructured light* method, featuring band-pass white noise patterns that are designed to be robust to indirect illumination by avoiding large black or white pattern regions.

In this paper, we consider surfaces that are mostly diffuse. If we can make one full period of our pattern smaller than the diffusion, then the effect of this diffusion is near constant for any pattern with the same frequency [50].

We limit the amplitude spectrum to a single octave, ranging from frequency f to $2f$, where a frequency refers to the number of cycles per frame. For each spatial frequency, the amplitude is set to 1 and the phase is randomized, subject to the

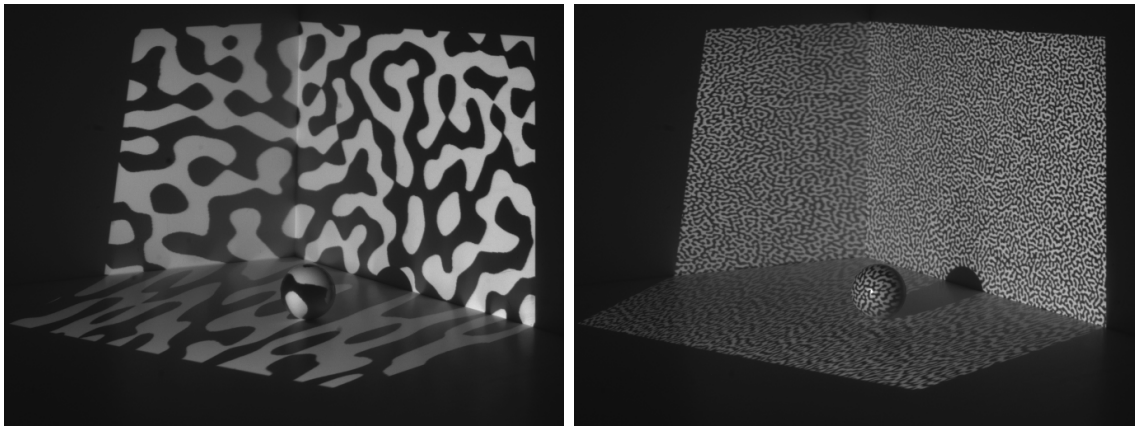


FIGURE 4.4: Synthetic patterns are generated in the Fourier domain by randomizing phase within an octave. Here, two patterns are shown projected on a scene. Spatial frequencies used are (left) 8 to 16 cycles per frame and (right) 64 to 128 cycles per frame.

conjugacy constraint [9], namely that $\hat{I}(f_x, f_y) = \overline{\hat{I}(-f_x, -f_y)}$.

The second step is to take the inverse 2D Fourier transform of $\hat{I}(f_x, f_y)$, yielding a periodic pattern image $I(x, y)$. To avoid periodicity, we generate a pattern larger than the desired width (say 110% larger) and then cut the extra borders. The pattern intensities are then rescaled to have values ranging in [0 :255]. Each pattern is finally binarized with a threshold at intensity 127 to make pixels either black ($\leq$ 127) or white ($>$ 127).

Hence, the patterns are parametrized by frequency f and limited to a single octave of variation to control the amount of spatial correlation (see Fig. 4.4). More spatial correlation increases code similarity locally, but also increases the number of required patterns to guarantee code uniqueness. We next discuss these two aspects.

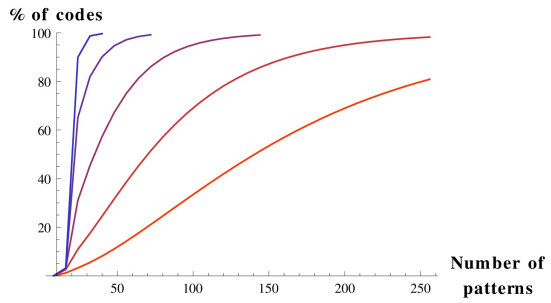
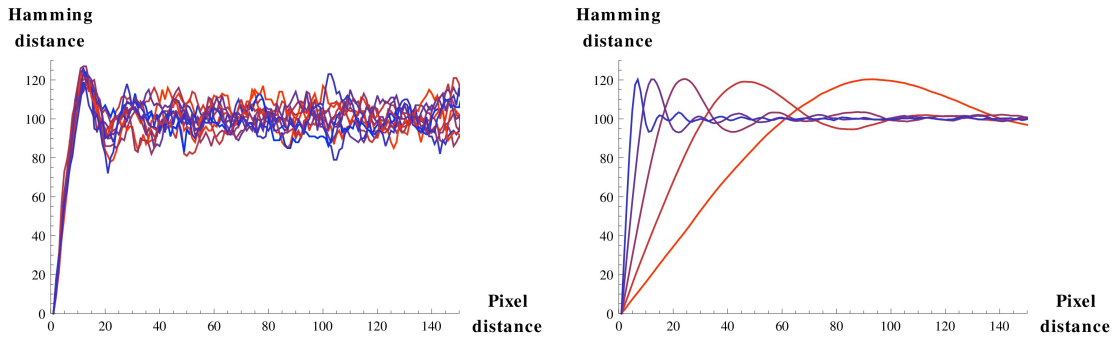


FIGURE 4.5: For HD images (1920 × 1080 pixel resolution), the percentage of pixels having unique codes while increasing the number of patterns. The curves correspond to f ranging from 8 to 128, with steep curves corresponding to patterns of higher frequencies. Curves stop being drawn if they reached 99%.



(a) (b)

FIGURE 4.6: Hamming distance between a randomly selected pixel and its neighbors with increasing distance, for a code length of 200 patterns. Distances are shown (a) for a few selected pixels ($f = 64$) (b) as the average over many selected pixels for different frequencies f ranging from 8 to 128, with steeper curves corresponding to patterns of higher frequencies. Each curve follows a sharp increase before decreasing to a constant that is half the number of patterns. Patterns of higher frequencies are not as correlated spatially (steeper increase).

4.4.1 Reducing code ambiguity

In this section, we analyze the relationship between frequency f and the number of patterns required to identify projector pixels uniquely with a code sequence of black and white values. Note that the pattern sequence is uncorrelated temporally to ensure that all bits in a code are independent.

In Fig. 4.5, we measure the number of patterns required to disambiguate at least 99% of all pixels as frequency f is varied. We consider HD projectors having 1920×1080 pixels. One can see that low frequency noise requires more patterns. Moreover, low frequency patterns often cause interreflections when large white pattern regions are projected in surface concavities and/or highly reflective materials.

Finally, we observed that this 1% of code duplicates usually correspond to small groups of neighboring pixels that have yet to be disambiguated. High frequency patterns, however, tend to quickly produce unique codes locally but have duplicates elsewhere.

One interesting question is whether 1D patterns could reduce considerably the number of required patterns. These 1D patterns could be used, for instance, in a calibrated setup for which epipolar geometry is known. Ignoring the fact that long 1D stripes do create more interreflection, using 1D stripes does reduce the number of patterns, but not considerably. The reason is that faraway codes usually get disambiguated after only a few patterns (in 1D or 2D), but local disambiguation takes a lot more pattern. For some fixed frequency, 1D disambiguation is faster than in 2D, but only by a factor or about 60% (data not shown). If two sets of 1D patterns are used (horizontally and vertically), then more patterns are actually required ($2 \times 60\%$).

4.4.2 Keeping neighbors similar

One important property of our patterns is the similarity between neighboring codes. Fig. 4.6 presents the hamming code difference with respect to the distance

between two neighboring pixels. Regardless of the frequency used, the hamming difference increases gradually with distance until it reaches a negatively correlated maximum before decreasing to a constant level. The standard deviation around this plateau is that of a Binomial distribution and is equal to about 7.07 bits, that is $\frac{\sqrt{N}}{2}$ for $N = 200$ bits.

This correlation between neighboring codes makes it easier for mismatch to happen between neighbors. However, it provides great robustness to pixel ratio variations, since the averaging of a group of neighboring codes is still highly correlated to each original blended codes. Also, this provides robustness to various local imaging problems like out of focus areas because of small depth of field.

Moreover, the lack of correlation between far pixels helps provide very high robustness to scene discontinuities. When a camera pixel observes a scene discontinuity, its intensity is a blend of two uncorrelated codes. Thus, about 50% of the bits are the same in both codes and will be accurately recovered. The remaining bits belong to either code, thereby ensuring that the matching code is composed of at least 75% of all bits of these two codes. This makes them and their neighbors much more likely to match than any other distant code. In contrast, if the recovered bits of two blended Gray code patterns are not all from the same code, then the resulting code may be completely unrelated to the two blended codes.

4.5 Establishing pixel correspondence

This section deals with efficiently establishing the correspondence between camera and projector pixels. We designed our matching method so that it does not require any form of prior calibration. By not using any epipolar constraint, matching becomes more difficult but much more flexible. For example, the camera could be a non single view point fisheye and the projector illumination could be bouncing off a convex surface. These cases are common in multi-projection setups and are not easily

calibrated [66].

A number of random unstructured light patterns are generated with a preselected band-pass frequency interval. Those patterns are projected one at a time while a camera observes the scene. N patterns are projected, captured by the camera, and then matched.

First, the gray images captured by the camera are converted into binary images for matching. The conversion is simply obtained by measuring if a pixel is above or below the average of previous patterns over time. Let $\Phi_{xy}(i)$ be a monotonic function modeling photometric distortion [3], the average image $\bar{I}_c$ in the camera, computed from all the distorted intensities in the camera, remains a good delimiter because it is well within $\Phi_{xy}(\text{black})$ and $\Phi_{xy}(\text{white})$ when, for a camera pixel, the amount of black and white values is reasonably balanced. Furthermore, the average works well because band-pass noise patterns should not produce big changes in indirect lighting.

Thus, as codes from *unstructured* light patterns no longer have any correlation to projector pixel position, pixel correspondences have to be found by matching two sets of high dimensional vectors to one another. Using N patterns, we obtain a N-dimensional binary vector for each pixel of both the camera and the projector image. For HD images, each set has around $1920 \times 1080 \approx 2$ million N-dimensional vectors. For the remainder of the section, we assume that camera pixels are matched to projector pixels, although matching can be performed the other way around (or even both ways simultaneously), which can be useful, for instance, in multi-projector systems [66] to remove the need to inverse the correspondence maps.

Efficient matching is achieved using a high-dimensional search method based on hashing of binary vectors as described in [3, 21, 4]. Algorithm 1 shows a pseudo-code of the matching algorithm. All vectors are hashed by selecting b-bits (hopefully noise free) out of the N code bits. We use a key size b that should cover at least the number

3. Photometric distortion includes gamma factors, scene albedo and aperture [11].

S of pixels in the projector such that expected number of codes hashed by a single key is around 1. In practice, we use $b = \lceil \log S \rceil$. While the codes should ideally match exactly (i.e. have the same key), there is some level of noise in practice. Thus, the method proceeds in k iterations, and selects a different set of bits for each iteration.

For a given pixel, the probability P that it is matched correctly after k iterations, in other words, that its hashing key has no bit error, can be modeled as

$$P = 1 - (1 - (1 - \rho)^b)^k \tag{4.1}$$

where ρ is the probability that one bit is erroneous. The number of iterations required to get a match within confidence P can be computed as

$$k = \frac{\log(1 - P)}{\log(1 - (1 - \rho)^b)} \ . \tag{4.2}$$

Several factors can increase the ρ value such as very low contrast and aliasing which becomes worse for higher frequency patterns and lower camera-projector pixel ratios. Thus, ρ can vary locally in the camera image, as scene albedo may change contrast for parts of the scene only. The pixel ratio may also change, in the presence of slanted surfaces for instance. Estimating ρ would yield an indicator of how many iterations are required, given the desired probability of a correct match P. However, Sec. 4.5.1 will introduce heuristics that improve convergence and thus, make the number of iterations predicted by ρ very pessimistic. Other termination criteria are discussed in Sec. 4.5.2.

Fig. 4.8(a) shows how adding code errors affects the convergence. We generated $N = 200$ patterns and applied a noise according to various ρ values. For instance, the best match should have an average optimal error of 20 bits for $\rho = 0.1$. One can see that convergence is still achieved for $\rho \leq 0.1$, but that it becomes much slower for higher ρ values. Since the number of iterations grows exponentially with ρ, a value larger than about 0.3 will result in no convergence.

Matching heuristics (see Sec. 4.5.1) can improve convergence considerably (see Fig. 4.8(b)). However, optimal matches do not guaranty quality matches. For instance, when $\rho = 0.3$ is used, the distribution of errors for good matches is not well separated from random codes ($\rho = 0.5$), distributed around half the number of bits $\frac{N}{2}$. We will discuss these distributions again in Sec. 4.5.1, in particular Fig. 4.9.

During an iteration, the hash table can be unbalanced, i.e. more that one code hashes in a single bin. The search for the closest code in each bin can increase significantly the matching time. In practice, the codes hashing to the same bin could be stored in a data structure accelerating the search. Instead, we select the first hashed code. Even if this strategy does not choose the best code, the time gained can be used to perform another matching iteration. Typically, the execution time for one iteration on a laptop with an Intel dual core 2.2 Ghz CPU with 2GB of RAM is around one second when matching an HD camera to an HD projector, and the iteration time is doubled when applying the heuristics.

Algorithm 1 Pseudo-code of the basic matching algorithm.

{assuming that the projector resolution is WxH}
$k \leftarrow \text{ceil}(\log(W * H))$ {compute the hashing key size for a hash table of size N}
$N \leftarrow 200$ {number of projected patterns}
for all camera pixels i **do**
  match_cost[i] $\leftarrow$ inf {init match costs to infinity}
end for
{keep matching until some criterias are met (see text)}
repeat
  mask $\leftarrow$ RandomMaskSelect(k, N) {select k bits out of N}
  proj_hash_table.init()
  for all projectors codes $P[i]$ **do**
    proj_hash $\leftarrow$ hash($P[i]$, mask)
    proj_hash_table.add(proj_hash, $P[i]$)
  end for
  for all camera codes $C[i]$ **do**
    cam_hash $\leftarrow$ hash($C[i]$, mask)
    $P[j] \leftarrow$ proj_hash_table.query(cam_hash) {closest projector code to C[i]}
    cost $\leftarrow$ HammingDistance($P[j], C[i]$)
    if cost $<$ match_cost[i] **then**
      match[i] $\leftarrow P[j]$
      match_cost[i] $\leftarrow$ cost
    end if
  end for
until some criteria is met

4.5.1 Matching heuristics

Usually, reconstruction methods take advantage of *a priori* knowledge about the scene in order to improve the results. One common assumption is that neighboring pixels have similar correspondences, thereby suggesting some form of local smoothing. Unfortunately, smoothing can introduce errors at discontinuities or wherever the assumption does not hold. In our case, we propose two simple heuristics that take advantage of scene smoothness to get a dramatic speedup in convergence. Their great advantage is that they improve the convergence time without any degradation of the final result.

The heuristics are illustrated in Fig. 4.7. *Forward matching* tests if a camera pixel can find a better match in the neighborhood of its current match in the projector. This heuristic refines matches that lie within the area of locally correlated region where cost increases with distance w.r.t. the best match (≤ 15 pixels in Fig. 4.6(a)). *Backward matching* tests the neighbors of a camera pixel to check if they could also match its corresponding projector pixel. This heuristic tends to create new matches, i.e. it improve current matches with potentially uncorrelated matches (> 15 pixels in Fig. 4.6(a)). The speedup is shown in Fig. 4.8, where the convergence is plotted as a function of the number of iterations needed with and without the use of the heuristics.

4.5.2 Match confidence and termination criteria

This section discusses a termination criteria to decide when to stop matching iterations. This is not a trivial problem due to the probabilistic nature of the algorithm. For instance, it can often happen that hashing improves a few matches even after there was no improvement for several iterations.

Camera pixels that see a surface area not directly illuminated by the projector should be excluded from the matching process because they produce random codes

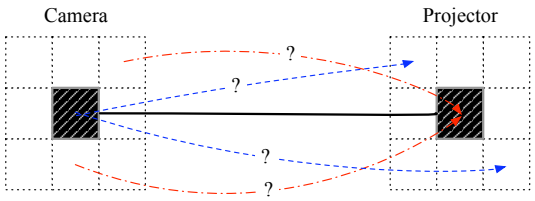
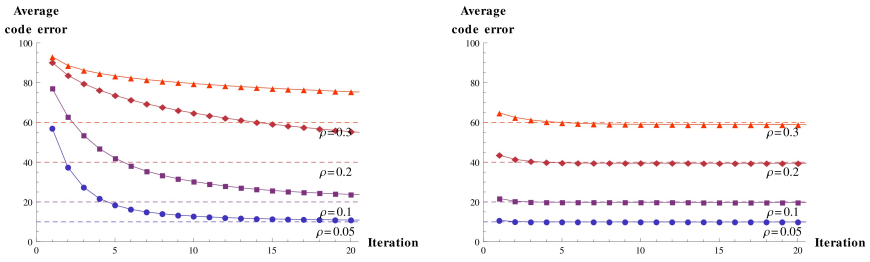
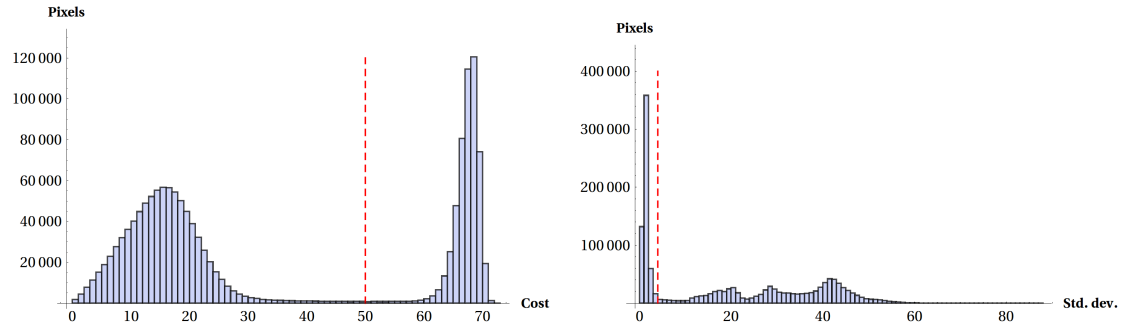


FIGURE 4.7: When a match is found (black solid line), two simple matching heuristics can be used : *forward matching* (blue dashed lines) attempts to improve an existing match and *backward matching* (red dot-dashed lines) attempts to create neighborhood matches.



(a) (b)

FIGURE 4.8: For increasing noise levels ρ, convergence of the hashing method (a) without heuristics (b) with heuristics. The dashed lines represent the theoretical lowest average code error. Convergence is much faster when applying the heuristics.



(a) (b)

FIGURE 4.9: For a typical scene, (a) a histogram of match costs has two distributions centered at ρN and at a value a bit below $\frac{N}{2}$ (see text for details). (b) a histogram of standard deviation of intensities has a high peak corresponding to unlit camera pixels or low contrast regions. A threshold (indicated here by the red dashed line) cannot completely separate the long tails of the distributions.

that depend on camera noise. The matching process would keep improving these matches, making a termination criteria more difficult to establish. Looking at the matching costs or standard deviations of intensity could be a good strategy to detect most of the unlit camera pixels. Fig.4.9(a) shows a histogram of the matching costs for a typical scene after 50 iterations. The matching costs are distributed in two well separated Binomial-like distributions, namely one centered at ρN and one centered below $\frac{N}{2}$ (in Fig.4.9, $N = 200$ and $\rho \approx 0.1$). The first distribution corresponds to correctly matched camera pixels. The second distribution corresponds to unlit pixels ; its mean is lower than $\frac{N}{2}$, because only the minimum matching code is kept at each iteration. Fig.4.9(b) shows a histogram of the standard deviations of pixel intensities. The distribution is roughly bimodal, with the highest peak corresponding to mostly unlit pixels. This narrow peak illustrates well the fact that all the patterns produce near constant indirect illumination for a given scene. Gray codes do not feature this property. The rest of the distribution is composed of lit pixels, modulated by the

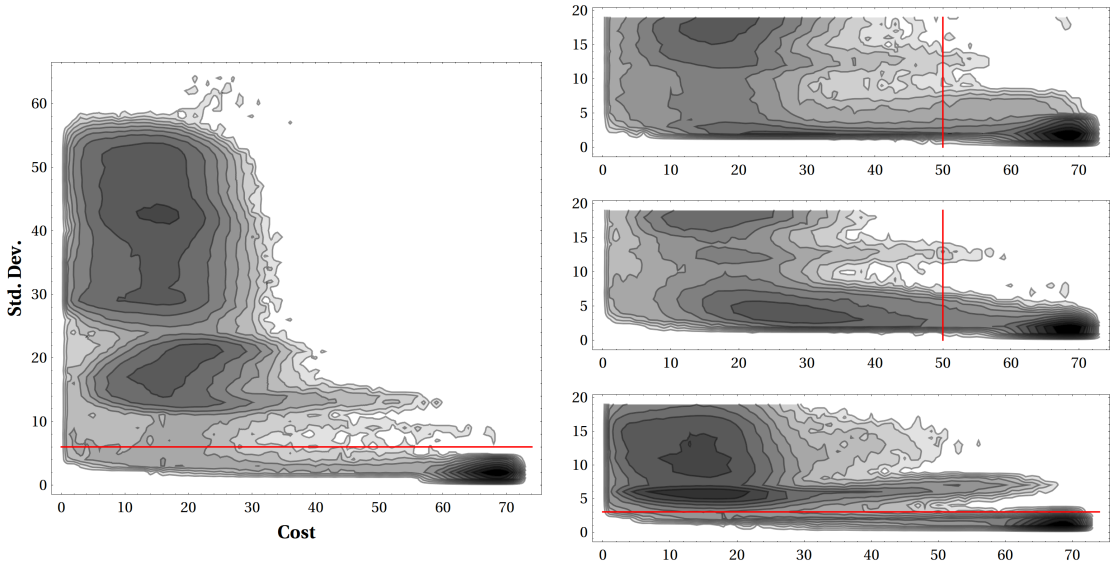


FIGURE 4.10: 2D log histograms of matching costs and standard deviations of intensity for the 4 scenes presented in the experimental results, namely (left) `Ball` and (right, top to bottom) `Games`, `Grapes & Peppers` and `Corner`. The red lines show the thresholds to remove unlit camera pixels.

scene reflectance.

However, this peak also contains pixels corresponding to dark scene objects. Because of this ambiguity, we consider both criteria, as illustrated in Fig. 4.10. Because of the long tails of the distributions, there is usually no single threshold which can separate all good matches from wrong matches. For most scenes, either criteria works. For scenes with dark objects, saturated or noisy imaging conditions, one criteria might work better than the other. The red lines illustrates the thresholds we used for the different scenes. In practice, both criteria could be used at the same time.

Once the unlit camera pixels are discarded, we can iterate until only a small number of pixels are updated (say 5 pixels) for a few iterations (say 5 iterations). Very few match errors may remain, usually less than 0.01% of all pixels (20 or 30 pixels). These are typically located where strong interreflection remains, such as the

intersection of two walls. There, the high code errors makes the heuristics inefficient. An exhaustive search is then performed for all matches that are not smooth with respect to their neighbors, in the hope of finding a better match. Smoothness for a camera pixel is simply checked by considering the average match of its neighbors, and verifying that it is within a threshold distance τ (we use τ=1.5). Note that this smoothness condition will also select all depth discontinuities as potential match errors, thereby subjecting them to an exhaustive search. This search is repeated until no further updates are made.

4.5.3 First results

In this section, we present the first results of our method on a real scene composed of two walls, a floor and a ball (see Fig. 4.1). The scene contains significant interreflections, depth discontinuities and out of focus regions. A more detailed comparison with other methods will be presented in Sec. 4.7.

Our method gives x and y correspondence map, as illustrated in Fig. 4.1. A frequency f of 128 cycles per image was used. Furthermore, we tested our method over a range of unstructured pattern frequencies. The results for selected regions are shown in Fig. 4.11. Notice that for regions not lighted directly, random codes are expected. This is observed behind the ball (Fig. 4.11 (top right)). High-frequency patterns also improve matching on the floor near the wall.

Finally, using the best results of our method as a reference, we measured errors by varying pattern frequencies and the number of patterns used. Fig. 4.12 shows that errors are smaller with more patterns and middle frequencies. Low frequencies are unsuitable to reduce the effects of indirect lighting, and more patterns are required to disambiguate codes locally. The fact that middle frequency patterns (here 32 and 64 cycles per frame) perform better than very high frequency patterns shows a tradeoff in the choice of frequency. While very high frequencies (here 256 cycles per frame) would be ideal to make indirect illumination near constant, they suffer from the

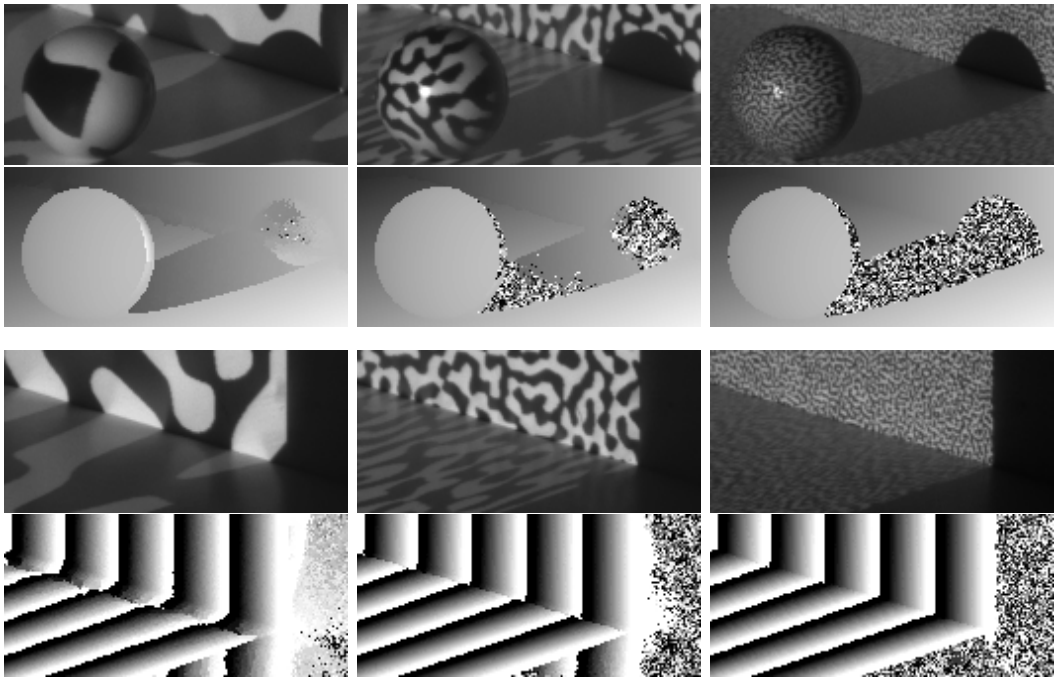


FIGURE 4.11: Correspondence from unstructured patterns at frequencies 8 (left), 32 (middle) and 128 (right). The effects of using higher frequency patterns are exposed on the edge of the ball and its shadow (top), and the corner of the walls (bottom).

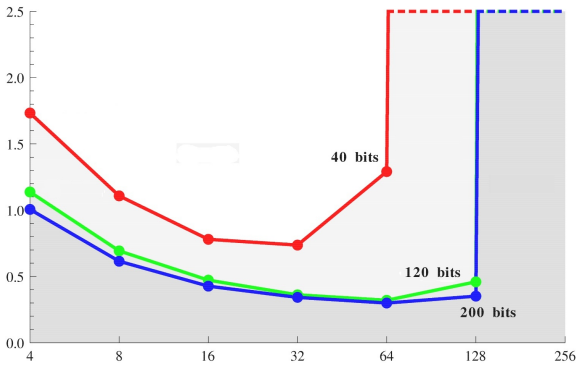


FIGURE 4.12: Average correspondence cost as a function of pattern frequency (4,8,...,256), for various code lengths (40,120 and 200 bits). Observe that more bits give lower errors. Low frequency patterns give slightly larger average errors because they required even more than 200 bits to disambiguate all pixels locally. High frequency patterns suffer from aliasing which makes convergence harder to achieve.

problem of camera aliasing, i.e. the camera resolution needs to be is sufficiently high to resolve the signal. They are also more prone to loss of SNR due to local blurring effects such as sub-surface scattering and defocus.

4.6 Comparison with the Gupta et al. *method*

This section compares our method to the method recently introduced in Gupta *et al.* [27] to address indirect illumination. Their method uses four set of codes, standard Gray codes and three other sets optimized for different illumination effects.

First, they address what they classify as long-range illumination (diffuse and specular interreflections) with the use of high-frequency patterns, generated by combining a chosen high-frequency base pattern with standard Gray codes through the XOR operation. From the captured images, the original Gray code patterns can be recovered by performing the XOR operation again with the same chosen pattern. Although this pattern could be any high-frequency pattern, Gupta *et al.* use the two highest Gray code patterns to generate two sets of patterns, namely XOR-2 and XOR-4 patterns (2 and 4 correspond to the maximum stripe width in both sets). Note that this choice produces narrow but very long stripes, which is not the case in our patterns. Effects of indirect illumination could probably be reduced further by choosing a base pattern that limits the stripes in both directions.

Second, they address short-range effects (sub-surface scattering and defocus) that can severely blur the high-frequency patterns, leading to a lot of code errors during the binarization process. To avoid this, Gupta *et al.* use a set of patterns called min-SW Gray codes [22], featuring stripe widths between 8 and 32 respectively.

Note that the XOR-2 and XOR-4 patterns do not maintain the basic Gray code property, namely that a code and any one of its neighbors differ only by one bit. This property ensures that if a camera pixel observes a mixture of two neighboring codes, then the dominant code is chosen from the one black/white transition (i.e. one bit

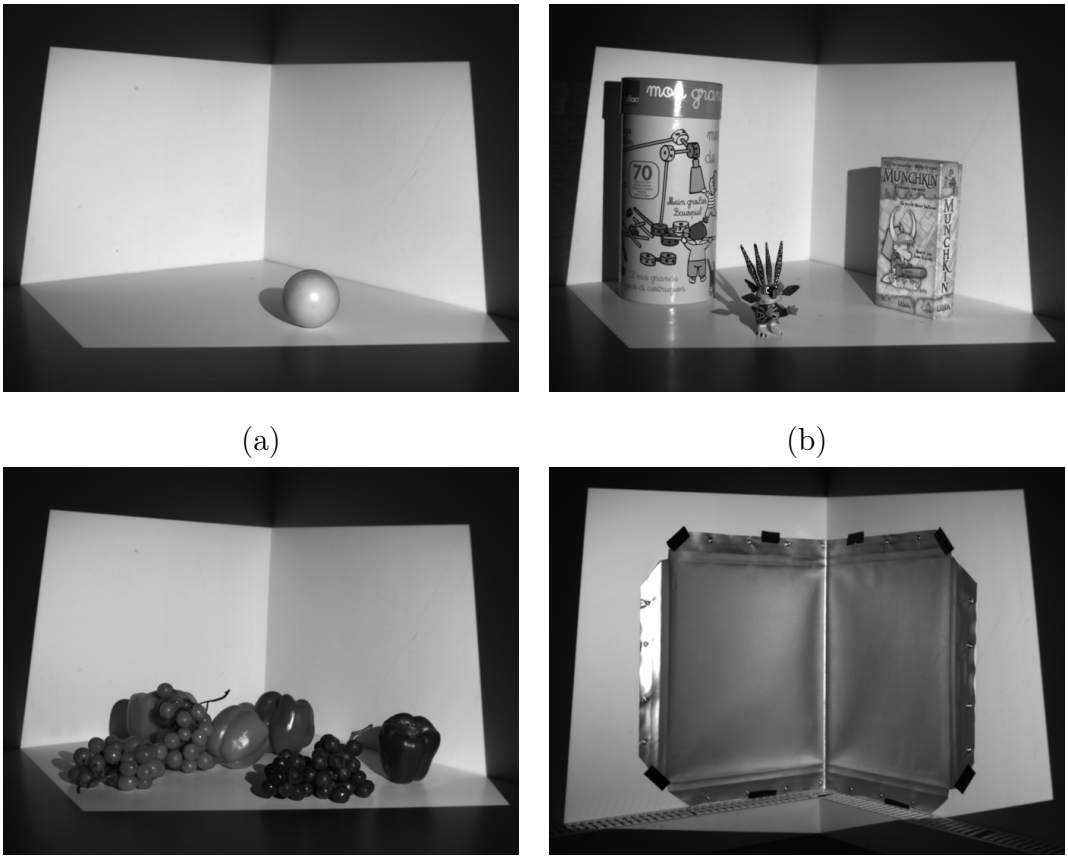
difference). But if more than one transition exists between two codes, then there is no guarantee that all dominant bits come from the same code, and the resulting code may then correspond to an unrelated far away pixel position. This is especially true in the presence of interreflections. In contrast, our method ensures local coherence.

In [27], good correspondences are chosen if they match in at least two sets of codes. Otherwise, a camera pixel is flagged as an error. In our implementation of the method, we matched codes in x and y separately and we considered that two matches agreed if their pixel distance was less or equal to 2. As in [28], we applied a 3×3 median filtering on the results to remove isolated noisy matches. Note that we did not address in this paper the iterative error correction process[76, 27] which captures additional patterns that include only unmatched projector pixels. While this process can be effective to decrease indirect illumination given a good error detection criteria, we argue that it should ideally not be required for robust patterns.

4.7 Experiments

In order to test the performance of our proposed method, we scanned several challenging scenes using a Gige Prosilica 1360 camera and a Samsung P400 projector. The pixel resolution of the camera and the projector were 1360×1024 and 800×600 respectively. We tested four scenes that exhibit different challenges : `Ball`, `Games`, `Grapes & Peppers` and `Corner` (see Fig. 4.13). We compared correspondence results from our method based on unstructured light (42 patterns), the Gupta *et al.* method using all 4 sets of patterns (42 patterns), the Gupta method using XOR-4 patterns only (12 patterns) and Phase-shift (3 patterns). Results using Gray codes and using our method with more patterns are available online at [1].

For our method, we used frequency $f = 64$, i.e. with frequencies ranging from 64 to 128 cycles per frame horizontally. We choose this range as it is about 4 times below the Nyquist frequency limit of 400 cycles per frame (the camera-projector pixel ratio



(a) (b)

(c) (d)

FIGURE 4.13: The four scenes that we tested, namely (a) `Ball`, (b) `Games`, (c) `Grapes & Peppers` and (d) `Corner`.

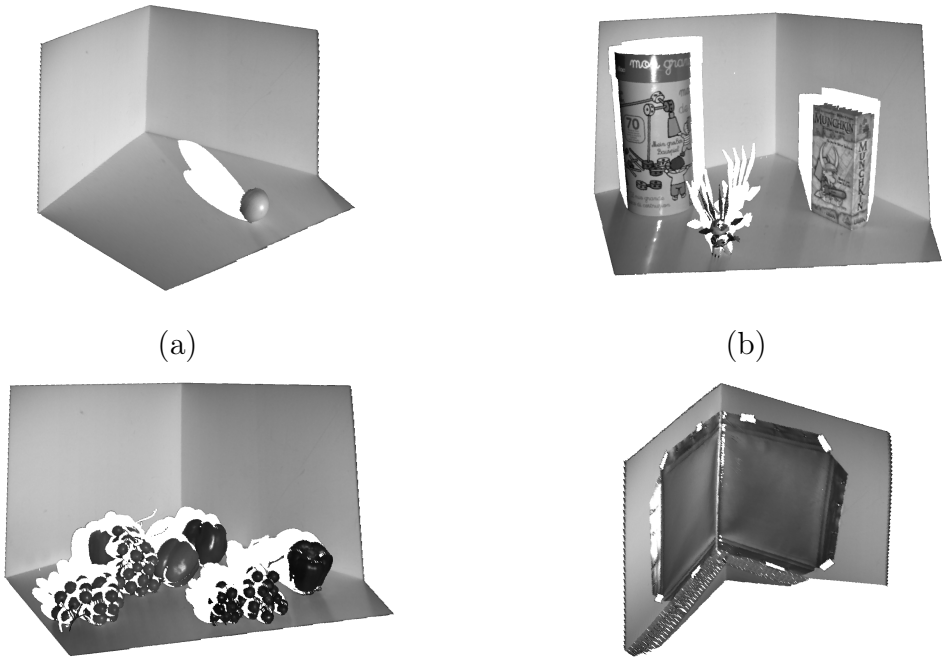
is approximatively 1) which adds robustness to out of focus regions and subsurface scattering.

The number of patterns required so that each projector pixel has a unique code is about 80. However, we used 42 patterns for a fairer comparison with the Gupta method. These still produce more than 99.9% projector pixels with unique code. The lower number of patterns make outlier matches more likely and so we apply a 3×3 median filter on the results. Note that our method finds x and y correspondence maps but that we only display x correspondences for comparison with the other methods. Generating 1D unstructured light patterns only would reduce the number of required patterns but we argue that the longer vertical strips create more indirect illumination than 2D patterns.

For each result, we computed the pixel difference with the unfiltered match given by our method using 200 patterns. For visualization purposes, the differences were scaled by 64, i.e. a 1-pixel difference has 64 pixel intensity, a 2-pixel difference has a 128 pixel difference, etc.

Ball

The `Ball` scene is similar to the scene used in Sec. 4.5.3. It is composed of two walls, a floor and a ball that creates a highlight and a depth discontinuity at its boundary. Results of all tested methods are shown in Fig. 4.15. Our method (top row) gives good results for high and low significant bits of the correspondence map. The Gupta *et al.* method (2^{nd} row) performs well, but a few pixels are discarded near the intersection of the wall and the ground, where interreflections are higher. Gray codes (3^{rd} row) fail to recover highly significant bits on the floor near the walls because of indirect lighting. Phase-shift (last row) results are presented for 16 and 64 cycles per frame patterns. Only low significant bits are shown (i.e. no phase unwrapping is applied). It has difficulties near the walls and features a wavy matching typical of direct coding methods in the presence of indirect lighting. This artifact gets worse when using a



(a) (b)

(c) (d)

FIGURE 4.14: Triangulation from the correspondences given by our method for (a) the `Ball` scene, (b) the `Games` scene, (c) the `Grapes & Peppers` scene and (d) the `Corner` scene.

lower frequency.

In order to verify that all methods perform similarly when unaffected by indirect illumination, we selected a region where indirect illumination is negligeable, namely the upper left region of the left wall, and compared the matches of all methods. At least 80% of the matches were exactly the same. All the remaining matches were within a distance of one pixel.

Games

Fig. 4.16 shows results for the `Games` scene, which exhibit a lot of sharp discontinuities. Also observe the curved surface of the cylindrical box, especially the soft edges at the sides where surface normals become perpendicular to the optical axis of the camera. There, Gray codes fail to recover correct matches. The Phase-shift method performs better, but the floor correspondences exhibit wavy results due to light bouncing off the cylindrical and rectangular boxes. The Gupta *et al.* method performs well, but error pixels are still flagged at the top of the rectangular box, the left and right edges of the cylindrical box and because of light reflection at its bottom. Our method successfully matches all these problematic areas. Notice that matches due to reflections at the left of the scene were not pruned because their matching cost was low even though contrast was low as well.

Grapes & Peppers

Results for the `Grapes & Peppers` scene are shown in Fig. 4.17. Grapes are translucent fruits that create subsurface scattering, and peppers have very shiny surfaces. Subsurface scattering is especially challenging to high-frequency patterns because they become blurry. Our method works quite well for this difficult scene. The Gupta *et al.* method fails to match pixels at the bottom of the right pepper and a few pixels on the grapes. Phase-shift and Gray codes also work pretty well, although Gray codes fail at the edges of the peppers, due to interreflections.

Corner

The `Corner` scene was made using two highly reflective surfaces set at a 90 degree angle. Gray codes and Phase-shift badly fail to match pixels near the corner. The Gupta *et al.* method is more successful in the sense that it does not exhibit wrong matches, but it misses alot of good matches. Our method works much better in that it is able to recover all matches, even at the corner. Notice that the black tape holding the reflective material could not be matched successfully because of its very low reflectance. The quality of the results of our method can be seen in Fig. 4.14 which shows all scenes reconstructed by triangulation.

4.8 Conclusion

In this paper, we addressed the problem of indirect illumination in structured light systems by taking advantage of a new approach to active reconstruction that uses patterns unrelated to projector pixel position. The only constraint imposed on these unstructured light patterns is that a sequence of these patterns identifies every projector pixels by a unique code. The proposed band-pass white noise patterns are designed to reduce the effects of indirect illumination and be robust to other issues such as low camera-projector pixel ratios. Because of the high number of patterns, the method is robust to capture errors and the matching algorithm provides very good performance with respect to depth discontinuities. Future works could address the problem of estimating matches at sub-pixel precision, as well as reducing the number of patterns by increasing the amount of new information given by each pattern, while still keeping their basic properties. It would also be interesting to investigate if the method could be used when multiple light sources are used [25] or when the projector is moving [30].

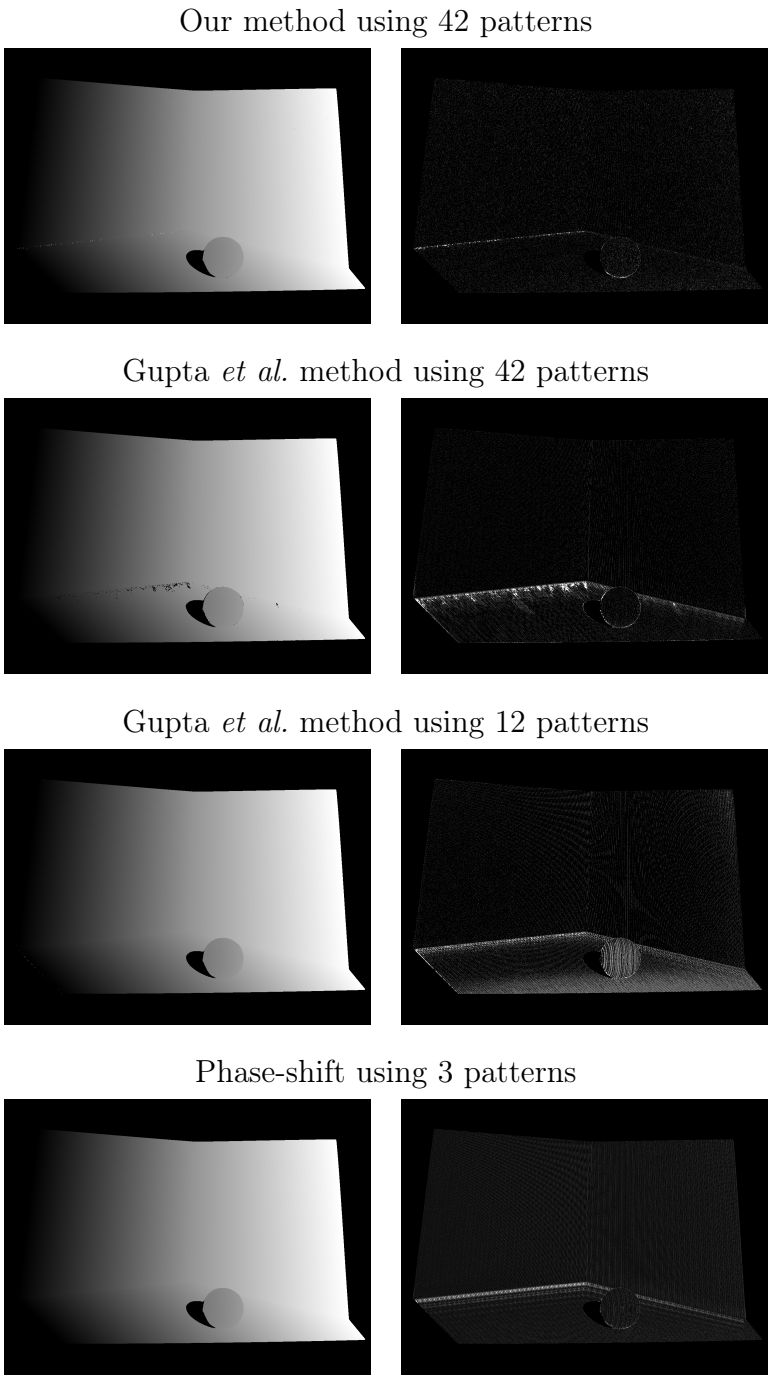
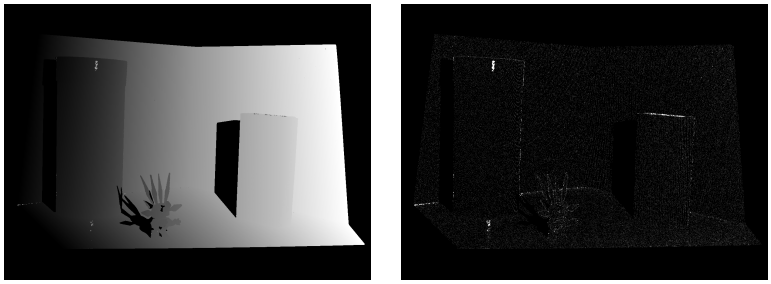
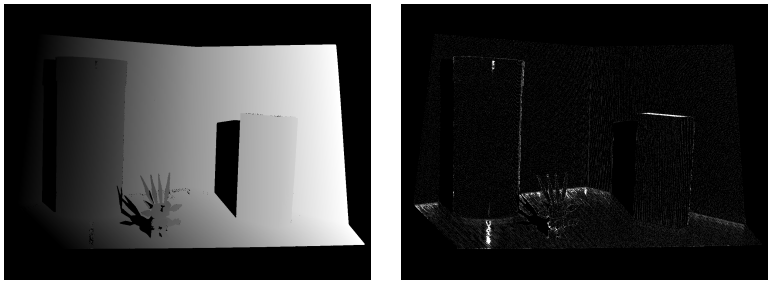


FIGURE 4.15: Results for the `Ball` scene. The left column show the x-correspondence map given by the tested methods. The right column shows the pixel difference w.r.t. the correspondences given by our method using 200 patterns.

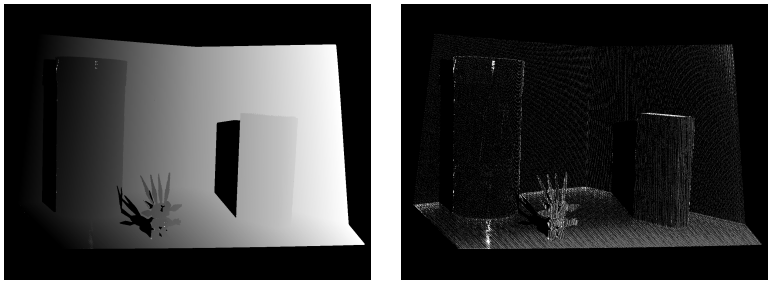
Our method using 42 patterns



Gupta *et al.* method using 42 patterns



Gupta *et al.* method using 12 patterns



Phase-shift using 3 patterns

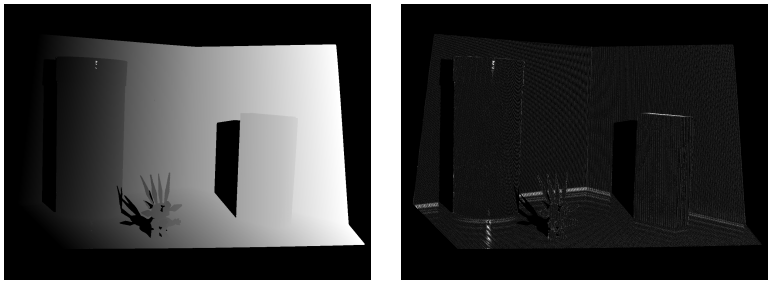
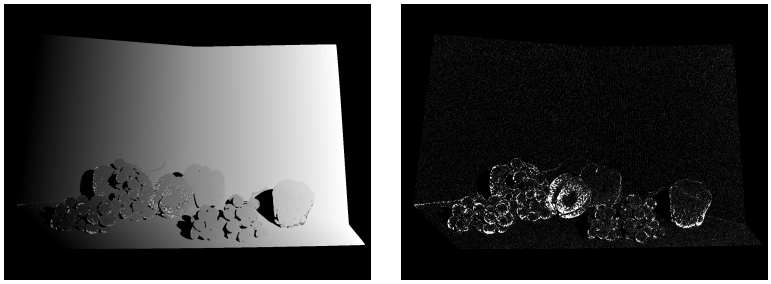
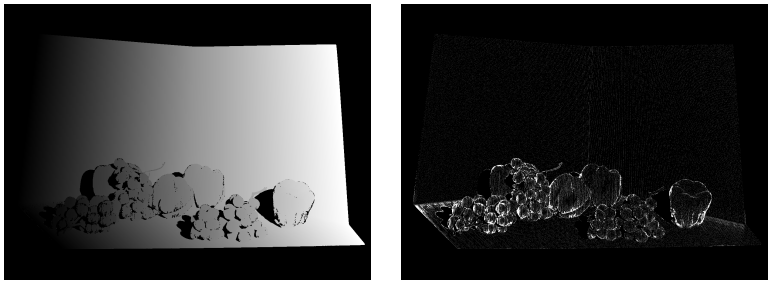


FIGURE 4.16: Results for the `Games` scene. The left column show the x-correspondence map given by the tested methods. The right column shows the pixel difference w.r.t. the correspondences given by our method using 200 patterns.

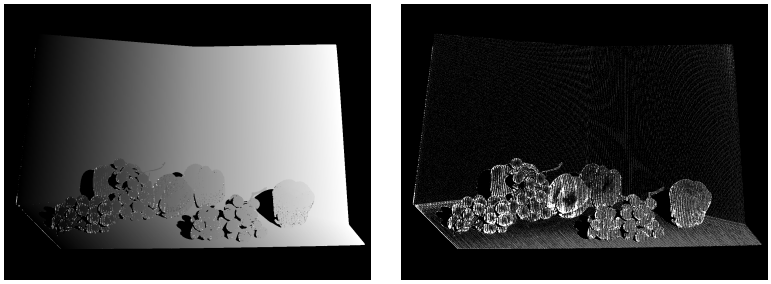
Our method using 42 patterns



Gupta *et al.* method using 42 patterns



Gupta *et al.* method using 12 patterns



Phase-shift using 3 patterns

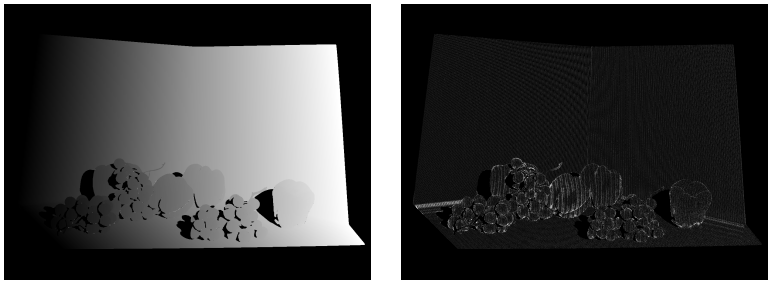
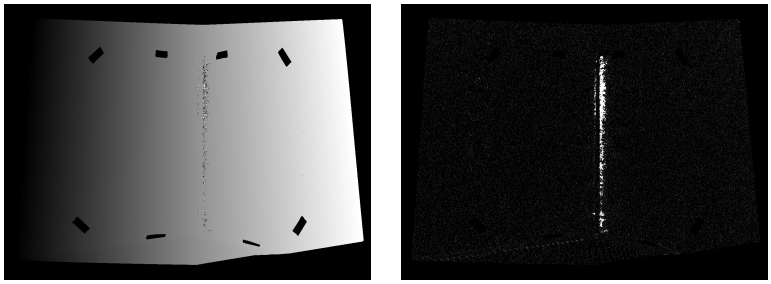
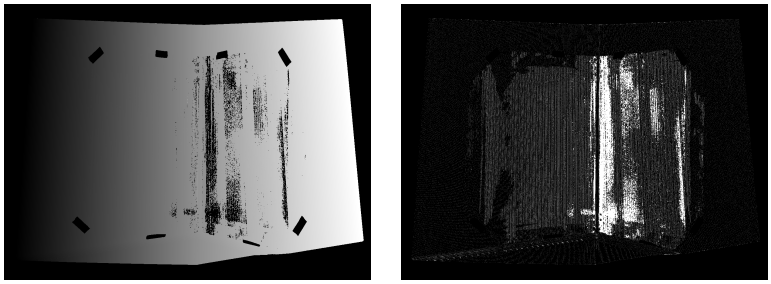


FIGURE 4.17: Results for the `Grapes & Peppers` scene. The left column show the x-correspondence map given by the tested methods. The right column shows the pixel difference w.r.t. the correspondences given by our method using 200 patterns.

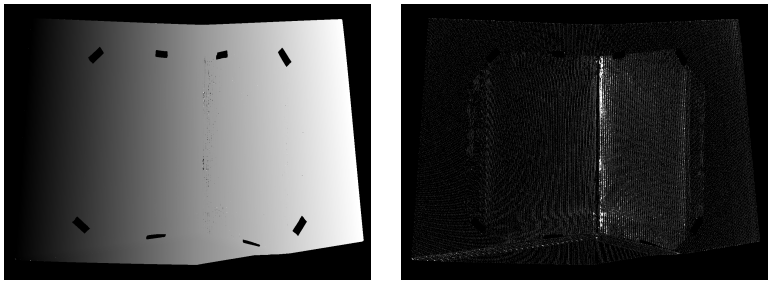
Our method using 42 patterns



Gupta *et al.* method using 42 patterns



Gupta *et al.* method using 12 patterns



Phase-shift using 3 patterns

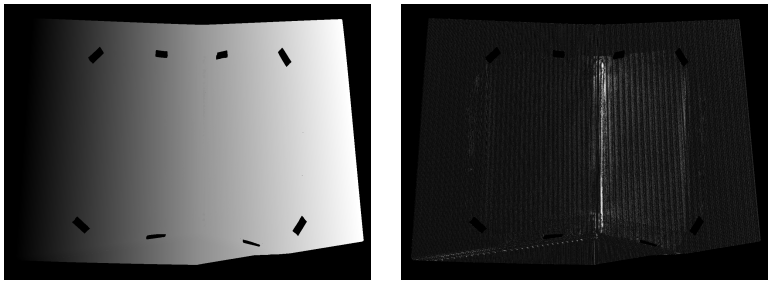


FIGURE 4.18: Results for the `Corner` scene. The left column show the x-correspondence map given by the tested methods. The right column shows the pixel difference w.r.t. the correspondences given by our method using 200 patterns.

Deuxième partie

Précision des méthodes de reconstruction active

Chapitre 5

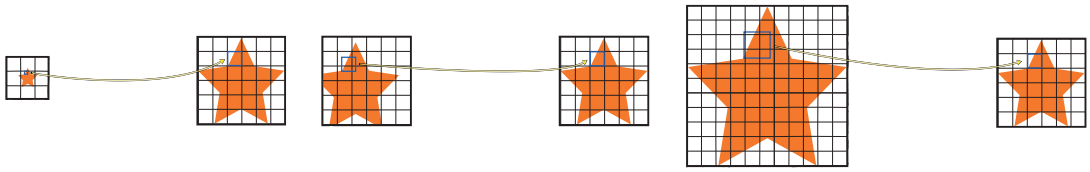
MISE EN CORRESPONDANCE SOUS-PIXEL

La précision de la reconstruction d'une surface 3D à partir d'images dépend de la capacité d'un algorithme à retrouver la position de la projection d'un point 3D de cette surface dans une image. Comme l'indique l'équation 1.2, il est clair que la position de la projection d'un point n'est pas une quantité entière. L'image d'une caméra étant une représentation discrète d'un signal lumineux, il est courant de parler de la correspondance entre deux pixels de deux images associés[60], mais c'est un abus de langage. En réalité, la correspondance d'un pixel d'une image se situe très souvent entre plusieurs pixels dans l'image prise d'un point de vue différent, ce qui justifie l'usage d'une représentation *sous-pixel* des positions correspondantes.

5.1 Ratio de pixels projecteur-caméra

Le projecteur et la caméra ont rarement la même résolution d'images. Par exemple, il n'est pas rare qu'une caméra possède une résolution de plusieurs mégapixels alors que le projecteur est de résolution beaucoup plus faible.

Le *ratio de pixels* est le ratio entre le nombre de pixels de caméra qu'il faut pour couvrir un pixel de projecteur. La figure 5.1 montre l'effet de la résolution sur le ratio de pixels entre la caméra et le projecteur. C'est une simplification, car la résolution n'est pas la seule variable à prendre en compte pour déterminer le ratio de pixel. En effet, même lorsque les résolutions sont similaires, la position relative de la caméra et du projecteur, ainsi que la disposition de la scène peuvent faire en sorte qu'un pixel de caméra ne "voit" pas la même portion de la scène qu'un pixel de projecteur, et donc que le ratio r soit différent de 1. Par exemple, c'est ce qu'on observe lorsque



(a) (b) (c)

FIGURE 5.1: Illustration de différents ratios r de pixels entre la caméra et le projecteur. Pour chaque figure, l'image de la caméra est à gauche et celle du projecteur est à droite. (a) ratio défavorable $r < 1$, (b) $r = 1$, (c) ratio favorable $r > 1$. Dans le cas (a), la correspondance n'est pas sous-pixel, à l'inverse des cas (b) et (c). Notons que dans (b), la correspondance est sous-pixel bien que les résolutions de la caméra et du projecteur soient les mêmes.

la projection se fait sur un objet plus proche de la caméra que du projecteur ou vice-versa. Remarquons que le ratio de pixel peut varier localement dans l'image, puisqu'il dépend de la profondeur des objets.

Il est important de noter que le ratio de pixel indique immédiatement si la correspondance entre la caméra et le projecteur est sous-pixel. En effet, si $r >= 1$ alors la correspondance entre la caméra et le projecteur est sous-pixel (voir figure 5.1-(b) et (c)). On parle de ratio défavorable ($r < 1$), lorsqu'un pixel de caméra couvre une surface beaucoup plus grande que celle d'un pixel de projecteur (voir figure 5.1-(a)). Dans ce cas, la correspondance entre le projecteur et la caméra est sous-pixel, alors que la correspondance entre la caméra et le projecteur ne l'est pas. Autrement dit, à un pixel de projecteur correspond une fraction d'un pixel de caméra, alors qu'un pixel de caméra couvre une surface comprenant plusieurs pixels de projecteur.

En résumé, si le ratio de pixel est défavorable, il n'y a aucun intérêt à obtenir une correspondance sous-pixels entre la caméra et le projecteur. Si la méthode le permet, on a plutôt intérêt à obtenir une correspondance entre le projecteur et la caméra. Dans tous les autres cas, il y a un gain à mesurer les correspondances avec

une précision plus grande que le pixel.

5.2 Nécessité des correspondances sous-pixels

Les reconstructions 3D obtenues à partir de correspondances entières entre la caméra et le projecteur souffrent d'un problème qui s'apparente au *crénelage*[73]. La triangulation à partir de coordonnées entières ne permet pas de retrouver des surfaces lisses, mais plutôt des surfaces en forme d'escaliers (comme le montre la figure 5.2). Certaines applications de la lumière codée pourraient s'accommoder de ce manque de précision, comme la déformation de la projection pour des surfaces planaires. Lorsque le but de la mise en correspondance est la reconstruction 3D, cette perte ne peut pas être négligée.

La figure 5.3 illustre le problème. Pour un pixel dans la caméra, illustré à droite par le point q, n'importe quel point le long de la ligne épipolaire correspondante, représentée par la droite orange dans le projecteur à gauche, pourrait être choisi comme correspondant. La vraie position sous-pixel est le point p dans le projecteur. En supposant que la méthode retrouve le bon pixel correspondant, n'importe quel point 3D situé sur le segment jaune pourra être obtenu après triangulation, à cause de l'imprécision de la correspondance. En l'absence d'information supplémentaire, le point 3D issu de la correspondance sera choisi afin de se reprojetter le plus près possible des centres des deux pixels. Lorsque la vraie correspondance se situe plus près d'un bord ou d'un autre pixel, le point 3D est alternativement estimé plus près ou plus loin que sa vraie valeur, ce qui explique l'effet escalier de la figure 5.2.

5.3 Estimation sous-pixel de la correspondance

Certaines méthodes de lumière codée calculent naturellement des correspondances sous pixels. C'est le cas des méthodes de déphasage de signaux sinusoïdaux qui obtiennent la phase d'un pixel à partir d'une intensité en ton de gris. D'autres méthodes

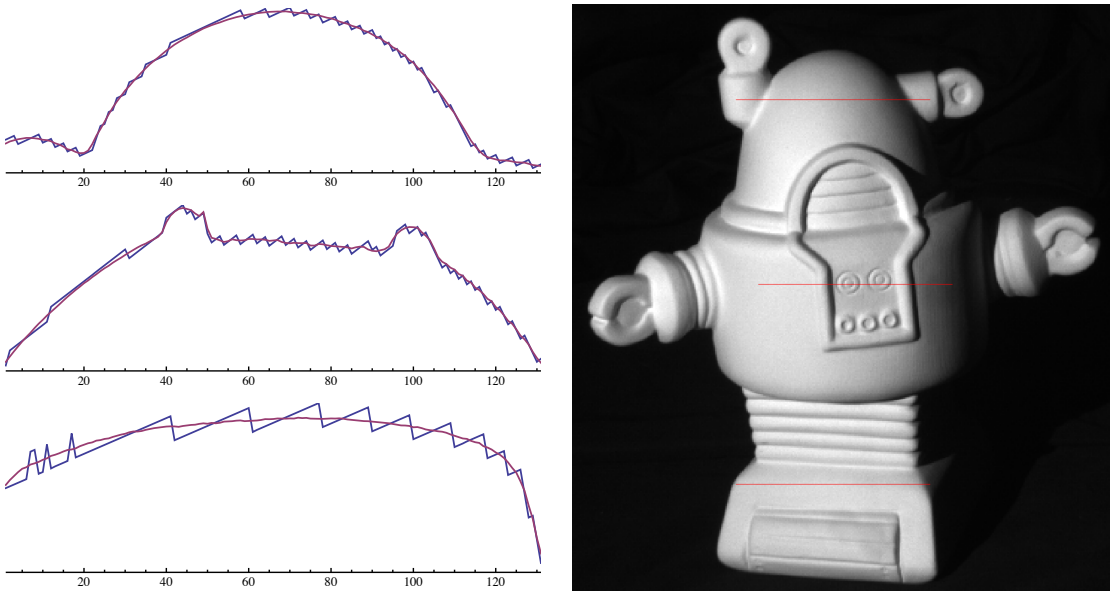


FIGURE 5.2: Reconstruction d'un objet utilisé dans le chapitre 6. À gauche, coupes en X de reconstructions 3D de l'objet. À droite, image de référence de la scène. Les lignes rouges correspondent aux positions des coupes présentées à gauche. Les courbes lisses (violettes) et en dent de scie (bleues) sont, respectivement, les reconstructions obtenues par notre méthode avec et sans sous-pixel.

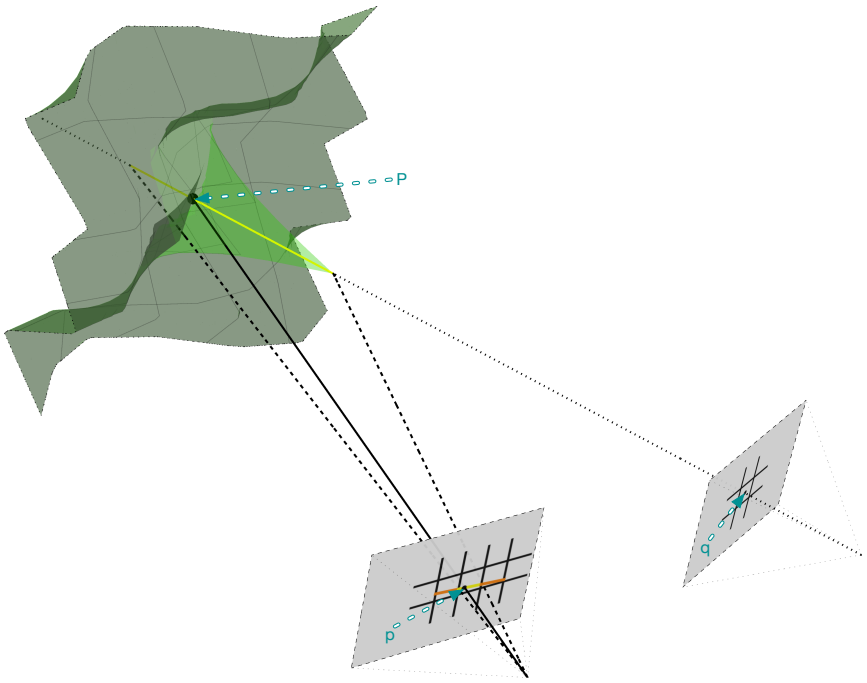


FIGURE 5.3: Incertitude d'une reconstruction non sous pixel. Le point P (inconnu) se projette en p dans le projecteur et en q dans la caméra. Si la méthode n'est pas sous-pixel, n'importe quel point 3D sur le segment jaune peut être reconstruit à partir de la correspondance entière entre p et q.

représentent uniquement les valeurs entières de la position d'un pixel à travers la séquence de motifs projetés, comme les méthodes à base de codes binaires. Des extensions ont été mises au point afin d'atteindre un niveau de précision supérieure dans le calcul des correspondances sous-pixels en déplaçant les motifs correspondants aux plus hautes fréquences[68] ou en projetant des motifs composés de lignes parallèles[67, 26].

Dans le cas des motifs de lumière non structurée, la méthode proposée au chapitre 4 calcule également des correspondances entières en construisant des vecteurs de bits extraits des pixels du motif projeté et de l'image capturée, et en cherchant ensuite les vecteurs les plus similaires dans chaque ensemble. Puisque chaque vecteur de bits n'est calculé qu'à des positions entières dans l'image du projecteur et de la caméra, il est nécessaire d'ajouter de l'information supplémentaire pour espérer extraire une correspondance sous-pixel.

Dans le chapitre suivant, nous modifions légèrement la procédure de génération des motifs de lumière non structurée et ajoutons une étape à la méthode de mise en correspondance afin d'extraire des correspondances sous-pixels. Dans un premier temps, il est nécessaire de générer des codes plus longs à partir des images projetées et capturées, afin de fournir suffisamment d'informations à l'algorithme de mise en correspondance sous pixel. À cette fin, nous construisons des motifs de lumière non structurée en ton de gris, plutôt que monochrome. Comme mentionné dans le chapitre 3, les vecteurs de bits calculés à partir d'images en ton de gris sont toujours invariants aux propriétés photométriques de la caméra et du projecteur puisque le signe de la différence entre deux intensités est invariant à ces paramètres. En revanche, les tons de gris permettent de former des vecteurs de bits beaucoup plus longs.

La méthode du chapitre précédent calculait le signe de la différence d'intensité d'un pixel avec la moyenne des intensités capturées pour celui-ci. Le chapitre suivant propose plutôt de calculer le signe de la différence d'intensité entre chaque paire

d'images pour un pixel. Avec des motifs binaires, cela n'aurait aucun intérêt, puisque du point de vue du projecteur une seule différence fournit toute l'information disponible, i.e. il est possible de déterminer l'intensité d'un pixel basé sur un seul signe puisque seulement deux valeurs d'illuminations sont possibles. Cependant, ce n'est pas le cas pour des motifs en ton de gris, et chaque paire d'intensités fournit *souvent* de l'information supplémentaire.

La méthode d'estimation sous pixel que nous proposons compare les codes de pixels voisins dans la caméra afin d'identifier la position sous-pixel la plus "probable". Il s'agit de la position qui, lorsqu'on effectue une combinaison linéaire des codes voisins, donnerait le code le plus similaire à celui du projecteur. Pour que cette méthode fonctionne bien, il est nécessaire d'obtenir des codes de longueur suffisante, donc projeter suffisamment de motifs, pour que l'estimation de la combinaison linéaire ne soit pas biaisée.

Chapitre 6

SUBPIXEL SCANNING INVARIANT TO INDIRECT LIGHTING USING QUADRATIC CODE LENGTH (ARTICLE)

Ce chapitre présente l'article[42] publié tel que l'indique la référence bibliographique :

> N. Martin, V. Couture et S. Roy, Subpixel scanning invariant to indirect lighting using quadratic code length, dans IEEE Computer Society International Conference on Computer Vision (ICCV)2013, IEEE, 2013, p. 1441–1448.

Dans cet article, nous proposons une méthode pour extraire davantage d'information des motifs de lumière non structurée. En effet, la méthode originale[15] exploite des codes de taille linéaire, c'est-à-dire que l'information disponible pour faire la mise en correspondance varie linéairement avec le nombre d'images projetées.

Les codes quadratiques présentés ici ont un avantage double. D'une part, ils requièrent la projection de beaucoup moins d'images pour obtenir les résultats de la méthode originale. D'autre part, les codes générés sont de dimension suffisamment élevée pour permettre l'estimation de correspondance sous-pixel qui améliore radicalement l'apparence d'une reconstruction 3D. Enfin, la robustesse de ces motifs à l'illumination indirecte ainsi qu'aux discontinuités de profondeur est maintenue, puisque les motifs, bien que différents des originaux, conservent la propriété "passe-bande".

Nous comparons les reconstructions obtenues à plusieurs méthodes sous-pixels, et en particulier à la méthode de micro déphasage [28].

L'article est présenté sous sa forme originale.

Abstract

We present a scanning method that recovers dense subpixel camera-projector correspondence without requiring any photometric calibration nor preliminary knowledge of their relative geometry. Subpixel accuracy is achieved by considering several zero-crossings defined by the difference between pairs of *unstructured* patterns. We use gray-level band-pass white noise patterns that increase robustness to indirect lighting and scene discontinuities. Simulated and experimental results show that our method recovers scene geometry with high subpixel precision, and that it can handle many challenges of active reconstruction systems. We compare our results to state of the art methods such as micro phase shifting and modulated phase shifting.

6.1 Introduction

Active scanning approaches using a camera and a projector have gained popularity in various 3D scene reconstruction systems [58, 57]. One or many known patterns are projected onto a scene, and a camera observes the deformation of these patterns to calculate surface information. Camera-projector correspondence is achieved by identifying each projector pixel by a code defined by the projected patterns.

The resolution of a projector being finite, several methods attempt to recover subpixel correspondences, thus giving better reconstruction results. In practice, it is often the case that a camera pixel observes a mixture of intensities from two or more projector pixels. The camera pixel integrates their intensities reflected from the scene, and the problem is then to estimate the composition of the measured intensity.

The main contribution of this paper is to present a method that recovers very high precision subpixel correspondence and is robust to indirect illumination. Our method uses a sequence of gray level band-pass white noise patterns to encode each projector pixel uniquely[15]. These are called *unstructured* patterns because the codes do not represent projector pixel position directly and a search is required to find the

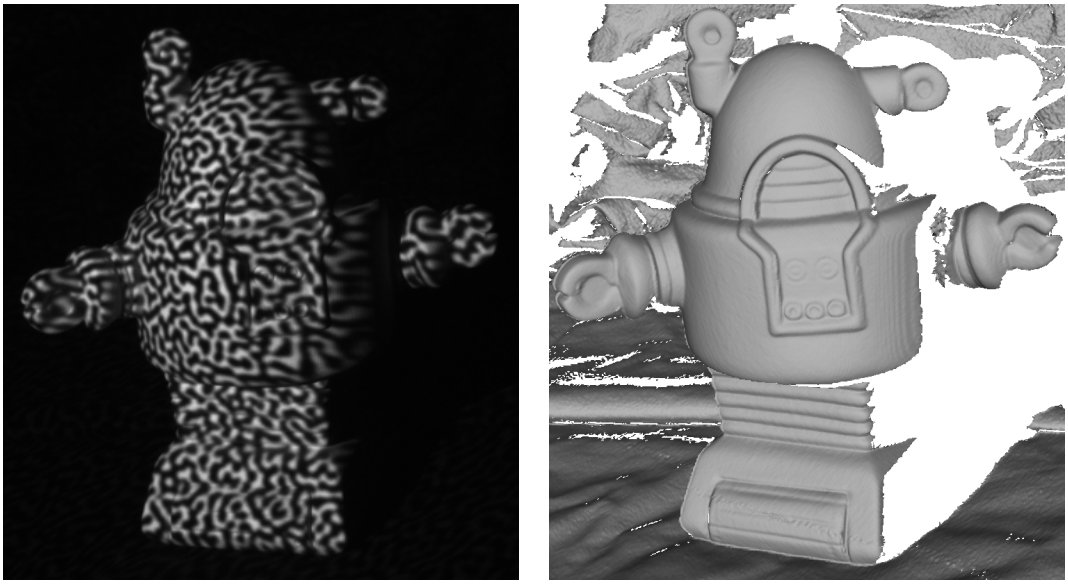


FIGURE 6.1: A band-pass gray level pattern projected on a scene (left) and its 3D reconstruction using our method (right).

best correspondence for each camera pixel [35, 17, 71, 15]. This approach is robust to challenging difficulties in active systems such as indirect illumination and scene discontinuities. Our method yields the same robustness as [15] while using a lot less patterns. Besides, it produces dense subpixel correspondence whereas the original method did not.

The key to achieving both subpixel correspondence and reducing the number of patterns is to increase the length of the code generated from the patterns. Instead of using the signed differences between each pattern and a reference as in [15], we consider differences between all possible pairs of blurred gray level unstructured patterns. The resulting codes are much longer than the number of patterns albeit with some redundancy. Every sign change between neighboring projector pixels provides a zero-crossing which is used as a constraint to recover subpixel correspondence. An example of our patterns is shown in Fig. 6.1 along with a 3D reconstruction.

The method we propose uses two-dimensional patterns and is designed to avoid

the need for geometric or photometric calibration of both the camera and the projector. While our method could rely on epipolar geometry to allow using one-dimensional patterns, we argue that they create more indirect lighting because of their low frequency in one direction [27]. Moreover, estimating epipolar geometry can in some cases be a tedious or impossible task. For example, it is nowadays quite common to use catadioptric or other non conformal cameras or projectors in multi-projector systems [66].

In Sec. 6.2, we summarize previous works in coded light systems, in particular to achieve subpixel precision. In Sec. 6.3, we introduce our method to increase the amount of information of unstructured light patterns. In Sec. 6.4, we show how to recover subpixel correspondence on synthetic data. We validate the method on real scenes in Sec. 6.5 and compare with results of state of the art methods. We conclude and propose future works in Sec. 6.6.

6.2 Previous work

The goal of this paper is to achieve a high precision subpixel reconstruction for static scenes in the presence of several challenges like indirect illumination, scene discontinuities or projector defocus (see [50] for a list of standard problems). Many active reconstruction methods can work at subpixel precision levels (see [58, 57] for extensive reviews). However, their accuracy is widely affected by their lack of robustness to the aforementioned difficulties[28, 15]. Some improvements were made possible lately by a careful redesign of the projected patterns[28, 13, 26, 27].

Several methods are based on the projection of sinusoidal patterns which encode the projection position by a unique phase [75, 79]. The pattern must be shifted several times and several frequencies are often needed [32]. A limited photometric calibration is required since the phase estimation is directly related to the intensities affected by the gamma of the projector. Modulated phase shifting [13] was introdu-

ced to generate less indirect illumination and increase the accuracy of the subpixel correspondences. The method modulates the highest frequency patterns with orthogonal high frequency sine waves. The number of projected patterns needed is very high however since each pattern is itself modulated by several shifted patterns. The method described in [25] can be used to reduce the required number of patterns by multiplexing the modulated patterns together. Due to the periodic nature of the pattern, all the above methods require a "phase unwrapping" step to disambiguate the phase recovered. Phase unwrapping involves lower frequency patterns that can introduce large errors [32], in particular because of indirect lighting [50]. Recently, micro phase shifting was introduced in [28] to unwrap the recovered phases using only high frequency patterns. Due to low frequencies in one direction, the projected patterns still produce some indirect illumination that can affect the results.

Another category of methods [35, 15] use so-called unstructured light patterns that form temporal codewords to identify each projector pixel uniquely, but require an explicit search to obtain correspondences. In [15], the patterns were designed to make constant the amount of indirect illumination, and the method was shown to be very robust. However, it did not yield subpixel accuracy reconstruction and required a lot of patterns.

6.3 From linear to quadratic code length

In [15], a camera pixel recovered a bit from the observed intensity by looking at the sign of the difference with the mean intensity over all patterns. The mean was considered a good reference because it is expected to be near constant when using a high enough frequency. For N patterns, a linear codeword $\dot{\mathcal{W}}$ of N bits is generated by comparing each captured pattern c_i with the average image $\bar{c}$ for each pixel $\mathbf{p} = (x, y)$. We have

$$\dot{\mathcal{W}}[\mathbf{p}] = \{\text{bit}(c_i[\mathbf{p}] - \bar{c}[\mathbf{p}]), 1 \leq i \leq N\} \tag{6.1}$$

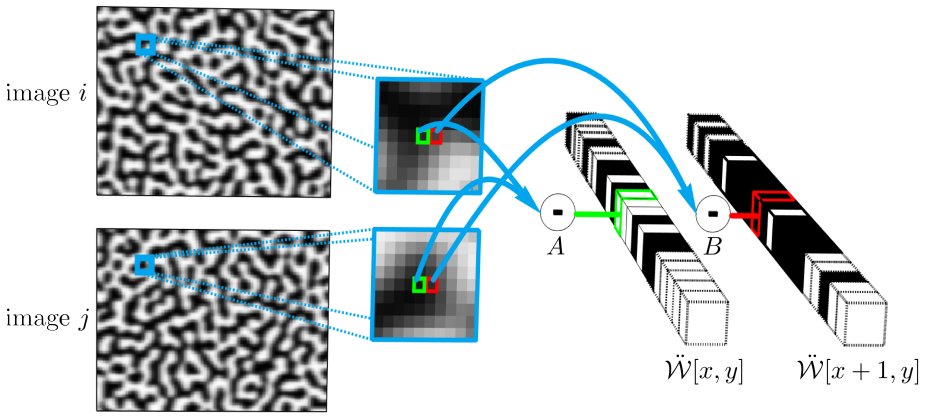


FIGURE 6.2: Bits are recovered by taking intensity differences between pairs of images. Two quadratic codes are shown for two adjacent pixels of the image pair (i,j). The labels A and B illustrates the computation of a bit of $\ddot{\mathcal{W}}[x,y]$ and $\ddot{\mathcal{W}}[x+1,y]$ as $\text{bit}(c_i[x,y]-c_j[x,y])$ and $\text{bit}(c_i[x+1,y]-c_j[x+1,y])$.

where $\text{bit}(a)$ has been defined as

$$\text{bit}(a) = \begin{cases} 0 & a < 0 \\ 1 & a > 0 \\ \text{random 0 or 1} & a = 0 \end{cases} . \tag{6.2}$$

We propose to increase the codelength by considering all possible pairs of pattern images as illustrated in Fig. 6.2. This provides a codeword $\ddot{\mathcal{W}}$ of quadratic length $\binom{n}{2}$ defined as

$$\ddot{\mathcal{W}}[\mathbf{p}] = \{\text{bit}(c_i[\mathbf{p}] - c_j[\mathbf{p}]), 1 \leq i \leq N, i < j \leq N\} . \tag{6.3}$$

This quadratic code is very unstable for binary patterns however, since half the intensity comparisons will yield differences of 0. We next explain how to generate our patterns which alleviate this problem.

6.3.1 Blurred gray level pattern generation

We propose to use band-pass gray level patterns which are generated as follows. Similarly to [15], we first apply a band-pass filter on white noise in the frequency domain, keeping only frequencies ranging from f to $2f$ where f is the same parameter as in [15]. After taking the inverse Fourier transform, the pattern is a random gray level signal composed of a limited range of frequencies. To produce uniform contrast across the whole pattern, we then binarize the pattern using a threshold at its average intensity, and then apply a blur kernel to make the pattern grayscale once again. The blur deviation should be close to $\frac{W}{6f}$ where W is the width of the image patterns, which is the average "radius" of black and white regions in our pattern (though the exact value used is not critical, see Sec. 6.4.4). In the next section, we analyse the number of patterns required to match.

6.3.2 Number of required patterns

Using these gray-level patterns, the quadratic code $\ddot{\mathcal{W}}$ now contains more information for each pixel than its linear counterpart $\dot{\mathcal{W}}$ but also some redundancy. The entropy of $\dot{\mathcal{W}}$ is clearly N bits. Since the entropy of N pairwise distincts elements is $\log_2(N!)$ bits, out of the $\frac{N^2-N}{2}$ bits of $\ddot{\mathcal{W}}$, only $\log_2(N!)$ actually provide information. As an example, 50 images will provide a quadratic code of length 1275 bits which effectively contains 214 bits of information. So a quadratic code from only 50 images is equivalent to a linear code of 214 images.

A minimum of 24 patterns is needed to uniquely encode each pixel of a 800×600 projector. This number could be slightly decreased if one allows the use of median filtering on the correspondence map (we do not advise this however, see our results in Sec. 6.5.2). The number of patterns is also expected to be lower when the epipolar geometry is known. Note that, in our experiments, we chose to use more than the minimal number of patterns to remove the number of images as a source of errors

and better assess the remaining reconstructions errors.

6.4 Achieving subpixel accuracy

As is the case with [15], the non-subpixel correspondence of a camera pixel is found using the LSH algorithm[3] that finds a match between the pixels of the camera and the projector, identified by the quadratic codes $\{\ddot{\mathcal{W}}^c\}$ and $\{\ddot{\mathcal{W}}^p\}$ respectively (using Eq. 6.3).

Assuming a camera-projector pixel ratio near 1, the camera pixel will generally see a mixture of four neighboring projector pixels. This mixture can be described by two parameters $(\hat{\lambda}_x, \hat{\lambda}_y)$ where $0 \leq \hat{\lambda}_x, \hat{\lambda}_y \leq 1$ which represent the subpixel matching disparity between camera pixel $\hat{\mathbf{p}}$ and projector pixel $\mathbf{p}$.

Consider that a camera pixel $\hat{\mathbf{p}} = (\hat{x}, \hat{y})$ has been matched to a projector pixel $\mathbf{p} = (x, y)$, using the LSH algorithm. To estimate $(\hat{\lambda}_x, \hat{\lambda}_y)$, we first need to find which quadrant represented by four projector pixels $\{(x, y), (x+\hat{\delta}_x, y), (x, y+\hat{\delta}_y), (x+\hat{\delta}_x, y+\hat{\delta}_y)\}$ adjacent to $\mathbf{p}$ out of the four possible, contains the sub-pixel match for $\hat{\mathbf{p}}$.

6.4.1 Selecting the right quadrant

There are four quadrants each composed of three projector pixels located around the matched projector pixel. The correct quadrant is selected as the pair $(\hat{\delta}_x, \hat{\delta}_y)$ for which the difference between the camera and projector codes is minimal :

$$\hat{\delta}_x, \hat{\delta}_y = \arg \min_{\delta_x, \delta_y \in \{-1,1\}} \Big(\Big| \ddot{\mathcal{W}}^c(x, y) - \ddot{\mathcal{W}}^p(x + \delta_x, y) \Big| + \Big| \ddot{\mathcal{W}}^c(x, y) - \ddot{\mathcal{W}}^p(x, y + \delta_y) \Big| + \Big| \ddot{\mathcal{W}}^c(x, y) - \ddot{\mathcal{W}}^p(x + \delta_x, y + \delta_y) \Big| \Big) .$$

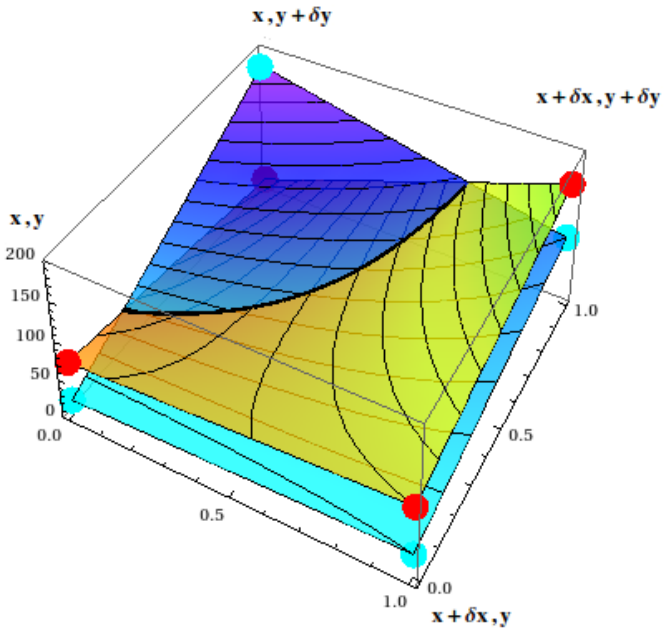


FIGURE 6.3: The red and cyan points corresponds to intensities of p_i and p_j respectively, for a quadrant out of four. The black curves represents the zero-crossing $S_{ij}(x, y, \delta_x, \delta_y, \lambda_x, \lambda_y) = 0$. Each pair (i, j) generates a 2D zero-crossing that provides constraints that are used to estimate the true subpixel position.

6.4.2 Estimating the subpixel position

For a projector pattern p_i, we model the interpolation of the intensities of the four neighboring projector pixels of a quadrant as a function of λ_x and λ_y using a bilinear plane :

$$\begin{aligned} K_i(x, y, \hat{\delta}_x, \hat{\delta}_y, \lambda_x, \lambda_y) = \\ \lambda_y\big(\lambda_x p_i[x, y] + (1 - \lambda_x) p_i[x + \hat{\delta}_x, y]\big) \\ +(1 - \lambda_y)\big(\lambda_x p_i[x, y + \hat{\delta}_y] + (1 - \lambda_x) p_i[x + \hat{\delta}_x, y + \hat{\delta}_y]\big). \end{aligned} \tag{6.4}$$

The 2D intersection of the two bilinear planes defined by projector patterns p_i and p_j is obtained by solving $K_i(x, y, \hat{\delta}_x, \hat{\delta}_y, \lambda_x, \lambda_y) = K_j(x, y, \hat{\delta}_x, \hat{\delta}_y, \lambda_x, \lambda_y)$. We define :

$$\begin{aligned} S_{ij}(x, y, \hat{\delta}_x, \hat{\delta}_y, \lambda_x, \lambda_y) = \; & K_i(x, y, \hat{\delta}_x, \hat{\delta}_y, \lambda_x, \lambda_y) - \\ & K_j(x, y, \hat{\delta}_x, \hat{\delta}_y, \lambda_x, \lambda_y) \\ = \; & A + B\lambda_x + C\lambda_y + D\lambda_x\lambda_y \end{aligned} \tag{6.5}$$

where

$$\begin{aligned} A &= p_j[x, y] - p_i[x, y] \\ B &= p_j[x + \hat{\delta}_x, y] - p_i[x + \hat{\delta}_x, y] - A \\ C &= p_j[x, y + \hat{\delta}_y] - p_i[x, y + \hat{\delta}_y] - A \\ D &= p_j[x + \hat{\delta}_x, y + \hat{\delta}_y] - p_i[x + \hat{\delta}_x, y + \hat{\delta}_y] - C - B + A. \end{aligned} \tag{6.6}$$

Equation 6.5 defines a polynomial which can be evaluated at any position (λ_x, λ_y) inside the region defined by the quadrant. The sign of the value gives the side of the curve $S_{ij}(x, y, \hat{\delta}_x, \hat{\delta}_y, \lambda_x, \lambda_y) = 0$ on which this point lies (see Fig. 6.3).

For each pair of patterns (p_i, p_j), the pair is discarded if the two planes do not intersect. Otherwise, if $\text{bit}(c_i[\hat{\mathbf{p}}] - c_j[\hat{\mathbf{p}}]) = \text{bit}(p_i[\mathbf{p}] - p_j[\mathbf{p}])$, then the subpixel position should be located on the side of the curve towards $\mathbf{p}$. Conversely, if the bits are

different, then it should be located on the other side of the curve. Thus, each pair (p_i, p_j) for which the planes intersect effectively provides a constraint on the value of the true subpixel location $(\hat{\lambda}_x, \hat{\lambda}_y)$. To account for the noise in camera codes, one cannot directly apply each constraint. In practice, $(\hat{\lambda}_x, \hat{\lambda}_y)$ should be voted as the value that satisfies the most constraints. In the next section, we present a hierarchical approach to efficiently solve this problem.

6.4.3 Hierarchical voting

The true subpixel position $(\hat{\lambda}_x, \hat{\lambda}_y)$ is the one satisfying the most constraints. It is found using a hierarchical voting scheme. At the highest level, the quadrant is divided into 4 equal square bins for $0 \leq \lambda_x, \lambda_y \leq 0.5$. Note that once the correct quadrant has been selected, the true subpixel location cannot be greater than 0.5 (otherwise the adjacent quadrant should have been selected). For each useful constraint, a bin gets voted if at least one of its corners is on the correct side of the curve. The process is then repeated recursively by dividing the winning square bin in four, until the desired amount of precision is reached. Note that if the two planes defined by a pair (p_i, p_j) do not intersect at some level, this constraint can be safely ignored at the next levels for more efficiency. For the experiments presented in this paper, we used 7 levels.

However, in practice, camera bits can have errors due to image noise, changes in surface albedo α and the gamma factor γ_p of the projector. We evaluate their effects in the next section.

6.4.4 Effects of noise and gamma

Image noise and several other factors lead to misleading constraints on $(\hat{\lambda}_x, \hat{\lambda}_y)$. We ignore in this paper the effects of scene albedo as we assume that it is constant within the field of view of a single camera pixel. Fig. 6.4 plots the RMS subpixel

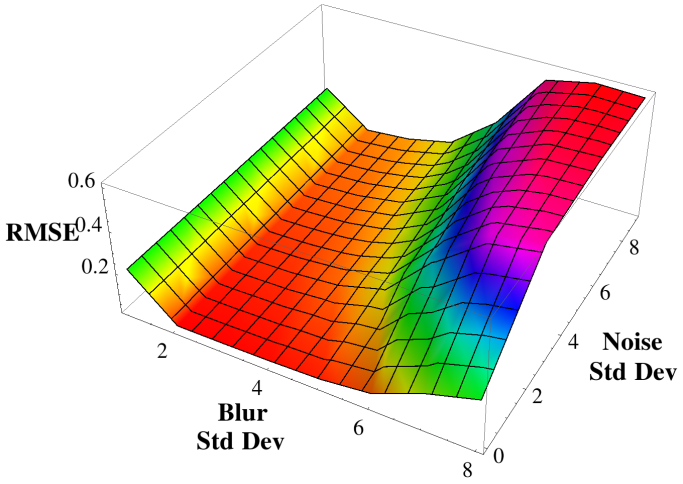
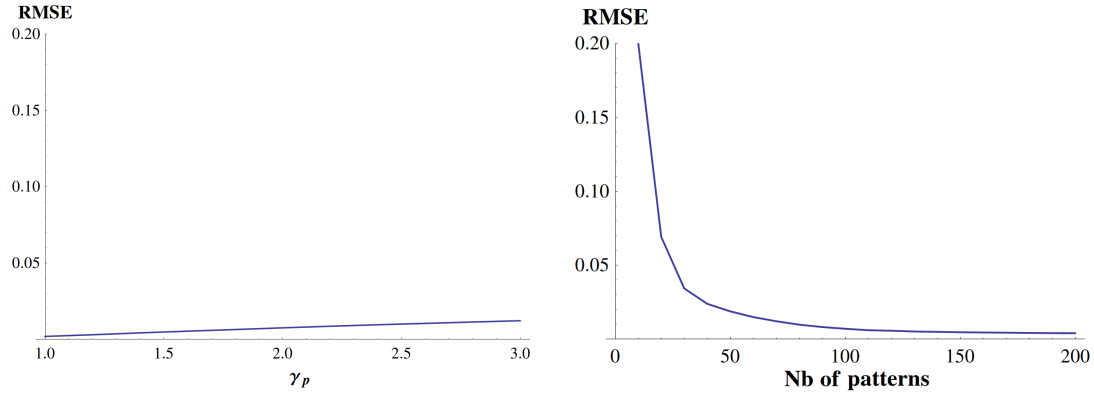


FIGURE 6.4: RMS subpixel error as a function of the standard deviations of the blur in pixels and the Gaussian intensity noise level.

error for different standard deviations of the blur in pixels and noise level. Synthetic subpixel positions were created by shifting 50 patterns of $f = 64$ cycles per frame and a 800×600 resolution. One can see that the exact blur deviation is not critical as there is a range going from 2 pixels to about 4 pixels that produce low error. In our experiments, we used a blur standard deviation of $\frac{800}{6*64} \approx 2$ (see Sec. 6.3.1). As for the gamma factor γ_p, Fig. 6.5(a) shows that its effect is very small when using 50 patterns. Finally, we also tested synthetically the error evolution when varying the number of patterns. Fig. 6.5(b) shows how the error decreases with the number of patterns. Note that, for all tests, we did not observe that the actual subpixel position has any effect on the RMSE (data not shown).

6.5 Experiments

In this section we describe the experimental setup we used to assess the quality of the reconstruction obtained with our method. We first present quantitative results with respect to an object for which the ground truth was acquired using an Arius3D



(a) (b)

FIGURE 6.5: For our synthetic experiment, the estimated subpixel location (a) is only slightly affected by the gamma nonlinearity of the camera. (b) is improved by increasing the number of patterns.

laser scanner. We then show various 3D reconstructions of a challenging scene and evaluate their quality by visual inspection. We compare our method to several other subpixel methods : the original phase shifting (PS) method of [79], modulated phase shifting (ModPS) presented in [13] and micro phase shifting (MicroPS) [28].

In all our experiments, we used a Samsung SP-400B projector with a resolution of 800×600 pixels and a Prosilica GC-450C camera with a resolution of 659×493. If needed by the method, a gamma correction was applied to the projected patterns. Each device was weakly calibrated independently and their final intrinsic parameters and relative geometry were found by bundle adjustment[2] for the purpose of 3D visualization. The set of points used for the minimization is the intersection of the correspondences estimated by each method so as to not introduce any bias toward a specific method in the subsequent 3D error measurements.

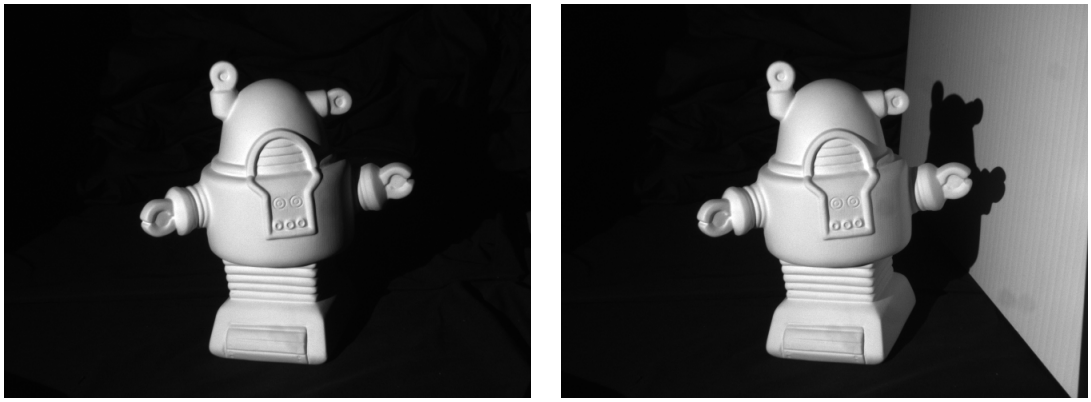


FIGURE 6.6: The robot was first scanned alone on a table (left), then a plastic board was added to the scene to create indirect lighting (right).

6.5.1 A simple scene with a ground truth : the robot

As shown in Fig. 6.1, a robot model was used in our experiments. To obtain a ground truth, it was scanned by Arius3D [1] at a very high resolution (0.1mm sampling, 10 micron RMS error).

In order to measure the sensibility of each reconstruction method to interreflection, we reconstructed the robot, with and without interreflections from a nearby plane (see Fig. 6.6). Fig. 6.7 illustrates how each method performed on a section of the robot model not affected by interreflections. Each method performed equally well, and the precision of the reconstruction is quite good. For fair comparison, every method used a budget of approximately 50 patterns to perform the scan. In order to do so, PS used 8 frequencies from 1/8 to 1/1024, 3 shifts per frequency for each direction (horizontal and vertical) for a total of 48 projected patterns. ModPS used 4 frequencies from 1/16 to 1/1024 : the highest frequency was modulated by 6 shifted versions of an orthogonal sinus wave of frequency 1/16 and each frequency used 3 shifts. The unwrapping used the method of [32] and 9 patterns per direction, so

1. `http://www.arius3d.com`

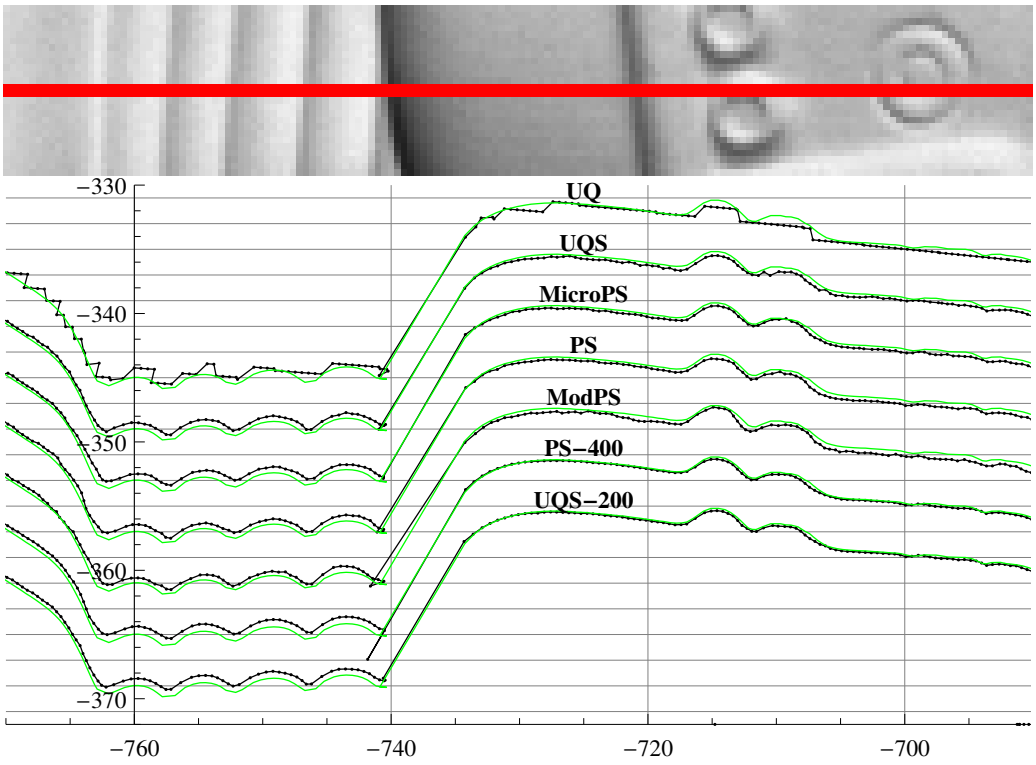


FIGURE 6.7: X-Z projection of reconstructed robot models for various methods. Units are in mm. The green curve is the reference scan. The portion of the robot which is reconstructed is illustrated in the cropped image at the top of the curves.

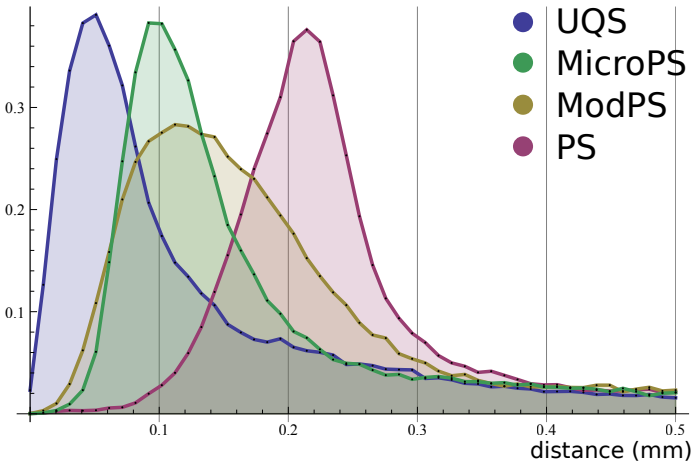


FIGURE 6.8: Histogram of reconstruction variations for the robot scene featuring strong interreflections.

ModPS used a total of 54 patterns. Finally, we slightly modified the original MicroPS method to use more than 3 shifted version of the highest frequency used to compute the wrapped phase. In [28], only 7 images were used. Since we wanted to use 50 patterns for all methods, 14 images were dedicated to unwrapping the phase in each direction (we used the frequencies recommended by the authors in the data available on their webpage [2]) and 11 shifted versions of a high frequency sine wave were used to compute the phase for a total of 50 patterns. We also added the results of UQ [15] which is not a subpixel reconstruction method but provides a reference to appreciate how well all the subpixel algorithms perform. PS-400 is the result of PS using 25 patterns per frequency (as opposed to 3 which is the minimum). UQS-200 is our method using 200 patterns.

We then evaluated the difference between 3D reconstructions and the ground truth in the area affected by interreflections. The histogram of variations is illustrated in Fig. 6.8. Our method (UQS) is the least affected, followed by MicroPS, ModPS

2. http://www.cs.columbia.edu/CAVE/projects/MicroPhaseShifting

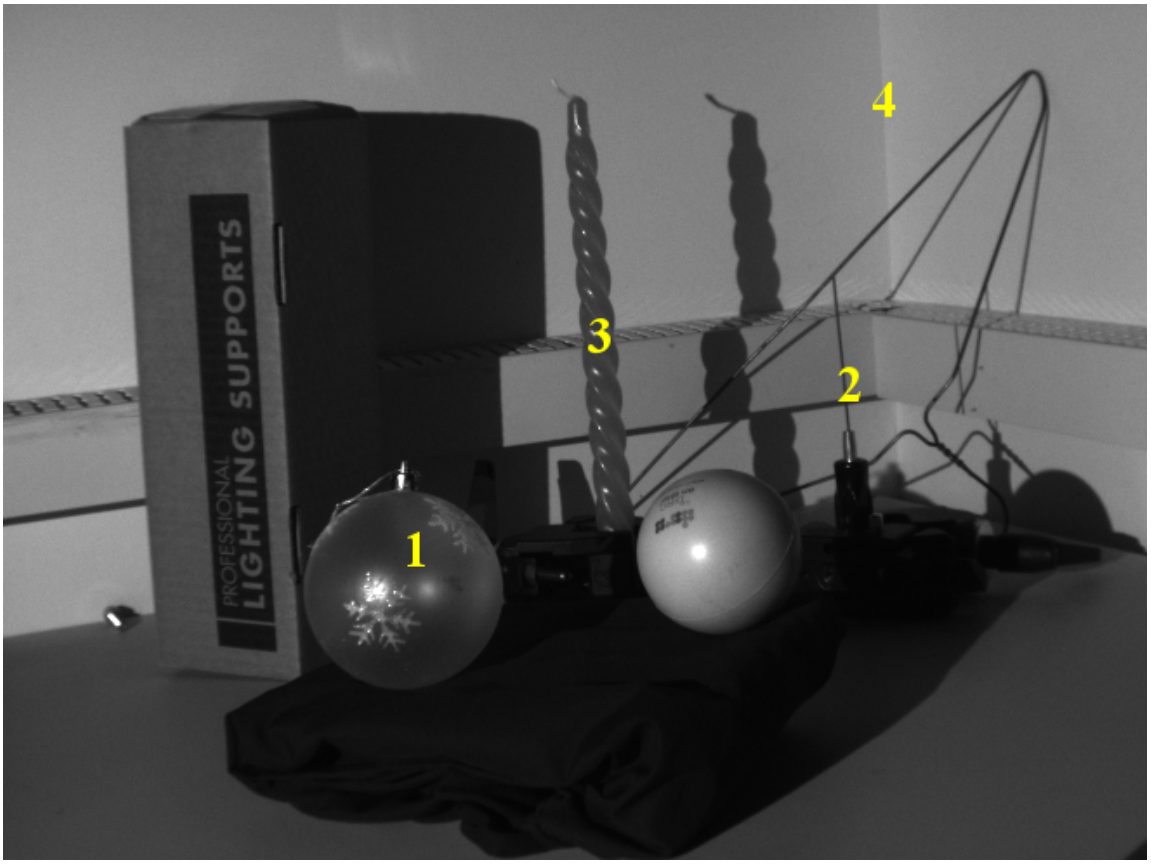


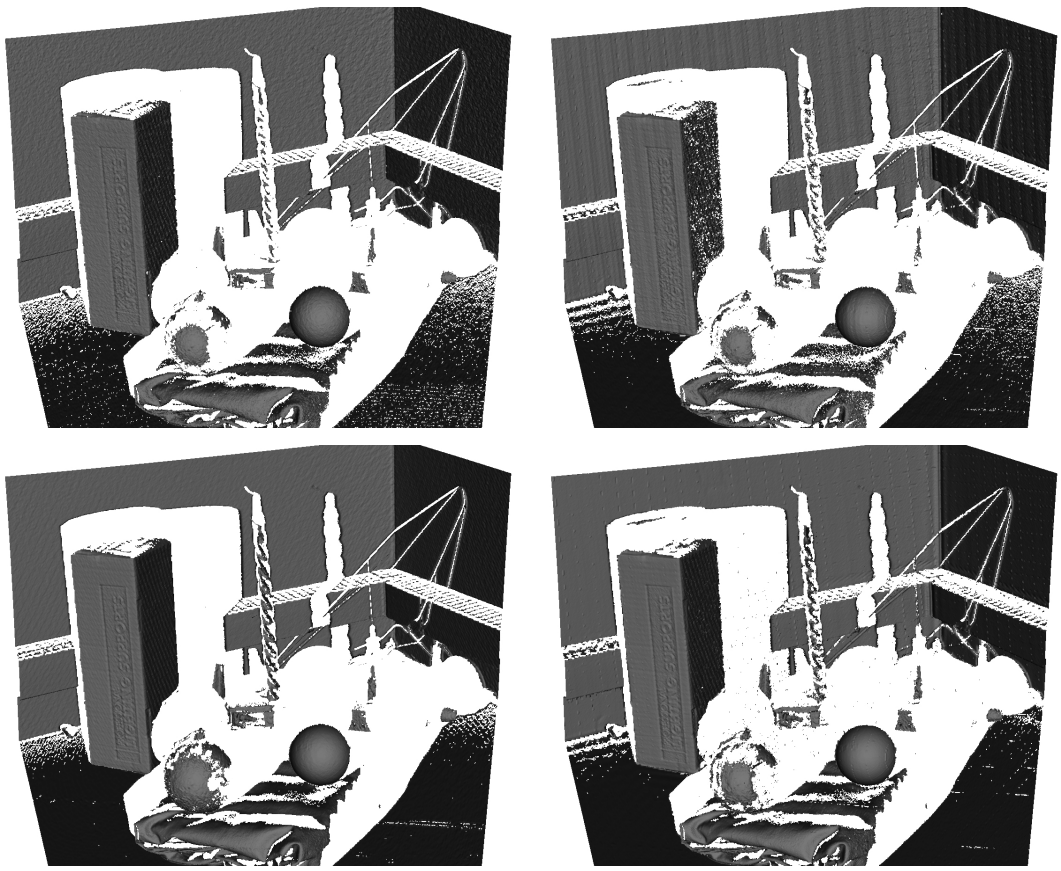
FIGURE 6.9: A complex scene featuring (1) translucency, (2) sharp discontinuities, (3) subsurface scattering and (4) interreflections.

and finally PS. It was expected that PS would be the worst performer since it does not feature robustness to interreflections. Overall, our results and those of MicroPS are very similar. They will be compared further on a more challenging scene in the following section.

6.5.2 Comparison with Micro Phase Shifting on a complex scene

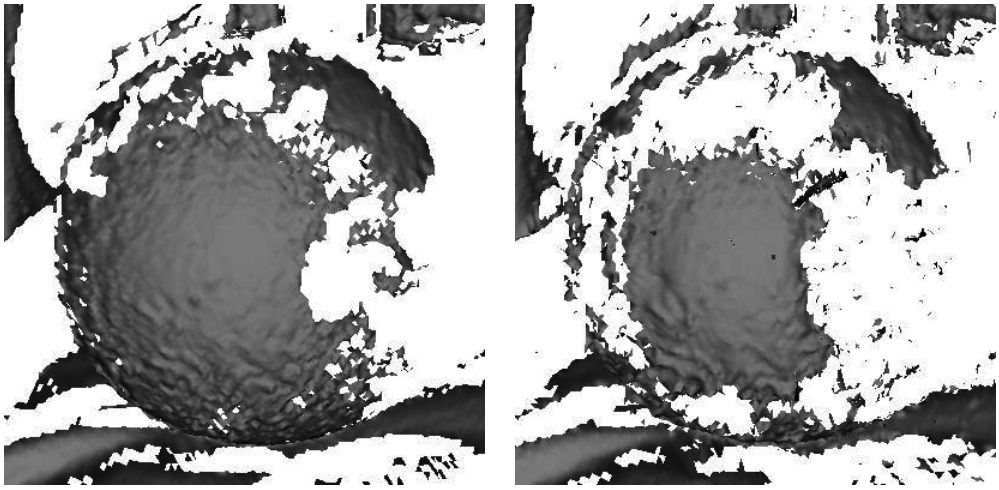
In this experiment, we scanned a scene composed of several objects which feature different materials and properties (see Fig. 6.9).

The 3D reconstructions we obtained with both methods are shown in Fig. 6.10.



(a) (b)

FIGURE 6.10: Reconstruction of a complex scene with (a) UQS (b) MicroPS. The number of patterns used was 50 (top row) and 200 (bottom row).

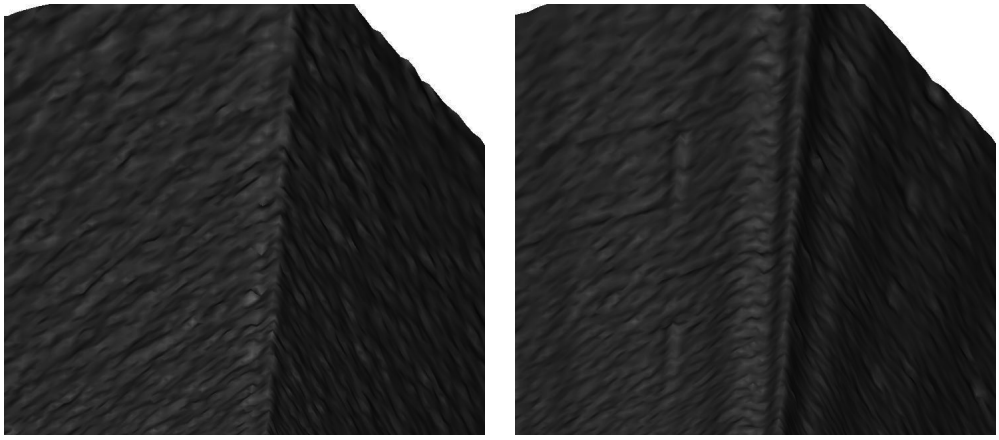


(a) (b)

FIGURE 6.11: Reconstruction of the translucent Christmas ball with (a) UQS (b) MicroPS. A larger portion of the ball is reconstructed using UQS.

The top row shows the reconstruction using 50 patterns for UQS, and 50 patterns for MicroPS. The bottom row shows the same reconstruction using 200 projected patterns for each method (for MicroPS, 86 patterns were used in each direction to estimate the phase). The reconstructions are similar for both methods at 50 patterns, even though some errors can be spotted in the reconstruction of slanted surfaces by MicroPS. It is however clear that UQS produces better results using 200 patterns. In particular, reconstruction was successful on a large region of the transparent Christmas ball, whereas MicroPS did not improve its results using 200 patterns, as shown in Fig. 6.11.

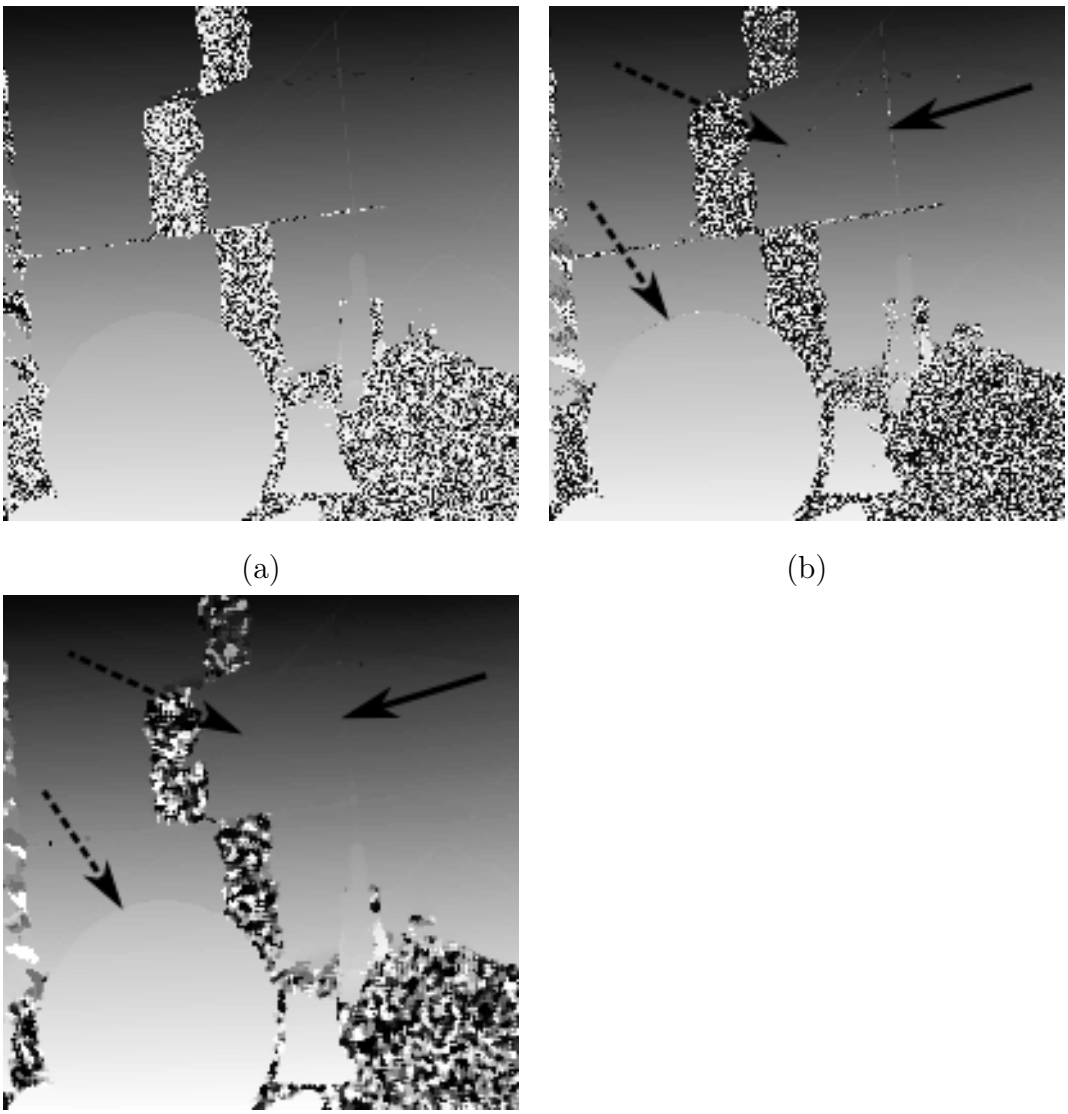
Note that the MicroPS method uses 1D high frequency patterns to unwrap and compute the phase. These generate more interreflections than our 2D patterns. This is especially visible in the corner at the back of the scene, where two bumps are falsely reconstructed as a result of some indirect lighting bouncing of each wall, as seen in Fig. 6.12.



(a) (b)

FIGURE 6.12: Reconstruction of the corner between the two walls with (a) UQS (b) MicroPS. Two bumps on each side of the corner are falsely reconstructed using MicroPS due to the indirect illumination generated by its 1D patterns.

Discontinuities can also be problematic for MicroPS. For instance, correspondences are erroneous on sharp edges or at the border of a discontinuity, as seen on Fig. 6.13. When using only 7 patterns as presented in [28], a median filter is applied to correct unwrapping errors and noisy phase estimates due to low signal to noise ratio. Since we used a lot more images, we found that the median filter was overall no longer necessary. However, when applied, the median filter does correct some errors (pixels on the edge of the ball for instance), but also removes the correspondences found on small objects like the screwdriver as shown in Fig. 6.13. MicroPS suffers from a trade-off between correspondence errors in discontinuities and the lack of correspondences on small objects. On the other hand, since our method does not rely on the use of a median filtering and naturally performs well in discontinuities, it does not feature this limitation.



(a) (b)

(c)

FIGURE 6.13: Correspondence maps of the screwdriver and hanger using (a) UQS (b) MicroPS without median filtering (c) MicroPS with 5x5 median filtering. Errors on the edges (dashed arrows) are present without median filtering, but sharp edges on small objects (plain arrow) disappear when applied.

6.6 Conclusion

We proposed a method to produce highly accurate subpixel correspondence using a projector and a camera. It relies on the principles of unstructured light scanning methods that are robust to common and challenging difficulties arising in active scanning systems. We use continuous gray scale patterns produced in the frequency domain. Subpixel position is estimated by comparing every pair of images and considering the location of zero-crossings. Each pair of images contributes a bit in quadratic codes that increase the information used in the subpixel estimation but also decreases the number of patterns needed to match. The method shown does not require knowledge of the epipolar geometry nor any photometric calibration. Reconstructions produced by our method were in general comparable to the ones produced by state of the art phase shifting methods, but showed increased robustness to indirect illumination and depth discontinuities.

Troisième partie

Comparaison des méthodes de reconstruction active

Chapitre 7

MOTIVATIONS D'UNE ÉTUDE COMPARATIVE DES MÉTHODES DE LUMIÈRE CODÉE

Les reconstructions obtenues à l'aide de motifs de lumière non structurée sont très précises, et résistent très bien aux défis standards des méthodes de lumière codée. Cependant, elles requièrent tout de même la projection d'un nombre important d'images pour fonctionner adéquatement, même lorsque la génération de code quadratique est utilisée (cf chapitre 6). Dans ce chapitre, nous abordons la question du compromis entre la qualité de la reconstruction et la quantité d'images utilisées par la méthode. En effet, certaines méthodes utilisent un nombre fixe d'images, mais d'autres peuvent améliorer leurs résultats en en utilisant davantage. Nous introduisons aussi des méthodes hybrides qui profitent des forces de plusieurs méthodes pour produire de bons résultats avec peu d'images.

Dans le prochain chapitre, nous comparons toutes les méthodes temporelles sur plusieurs plans. Nous évaluons la robustesse et l'exactitude de chaque méthode dans des conditions difficiles, et en particulier lorsqu'un nombre limité d'images est utilisé. Pour effectuer une évaluation équitable, nous argumentons que les correspondances seulement devraient être comparées, et non les reconstructions 3D comme c'est la pratique habituelle dans le domaine. En effet, les reconstructions 3D encapsulent des erreurs externes à la méthode de correspondance, comme les erreurs de calibration, qui peuvent biaiser les comparaisons. Nous utilisons donc une méthode spécialisée dans la capture des interactions lumineuses entre le projecteur et la caméra afin de calculer une correspondance “optimale" à des fins de comparaison.

Pour illustrer le compromis entre l'exactitude de la reconstruction obtenue et le nombre d'images utilisées, nous introduisons le concept de dualité quantité/qualité.

Ensuite, dans l'optique de présenter notre méthode d'acquisition d'une carte de correspondance de référence, nous discutons de la *matrice d'illumination* (*light transport matrix*) qui est normalement utilisée pour générer des motifs de projection aux propriétés optimales du point de vue de la caméra[7]. Cette matrice encapsule les propriétés géométriques et photométriques de la caméra, du projecteur et de la scène. La méthode de calcul des correspondances "optimales" que nous présenterons au prochain chapitre s'en inspire en grande partie.

7.1 Dualité quantité-qualité

Bien que les méthodes temporelles ne soient pas, à l'origine, directement applicables au temps réel, réduire le nombre de motifs à projeter est toujours souhaitable. D'ailleurs, il est possible, dans certaines conditions, d'obtenir des reconstructions que l'on pourrait qualifier de quasi-temps réel[49, 79, 28]. La méthode que nous avons utilisée au chapitre 6 permet de réduire le nombre de motifs, mais telle que publiée, elle requière toujours autour de 50 images. En pratique, nous avons obtenu de très bonnes reconstructions avec aussi peu que 25 images, mais ce nombre dépend beaucoup de la scène observée. En définitive, utiliser moins de motifs oblige à faire des sacrifices au niveau de la qualité des correspondances récupérées et ultimement au niveau de la reconstruction.

Une façon standard de réduire le nombre de motifs à projeter de moitié, est d'utiliser la géométrie épipolaire. Cela implique que la caméra et le projecteur soient préalablement calibrés. Par exemple, les codes de Gray ou les méthodes de déphasage peuvent ne projeter que les motifs correspondants aux codes dont la direction est perpendiculaire au "baseline" (voir figure 7.1). Ces méthodes utilisent des codes séparables, c'est-à-dire que le motif associé à un axe (X ou Y) est répété partout le long de l'autre axe. Il est possible de ne projeter que la moitié de tels codes si la géométrie épipolaire est connue, car celle-ci détermine déjà une dimension de la loca-

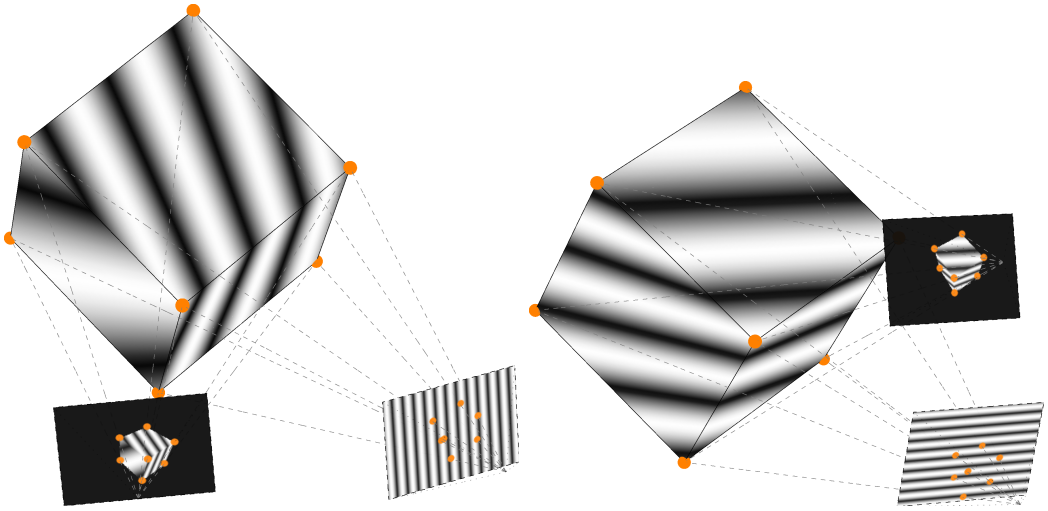


FIGURE 7.1: Lorsque la géométrie épipolaire est connue et que les motifs sont *séparables*, il suffit de projeter les codes dans la direction orthogonale au “baseline". À gauche, la géométrie épipolaire a une composante majoritairement horizontale et les motifs sont orientés verticalement, et inversement à droite.

lisation du pixel avec la ligne épipolaire (voir la sous-section 1.3.3). Bien entendu, les motifs choisis doivent être compatibles avec la géométrie épipolaire pour constituer une correspondance complète.

Les motifs basés sur la lumière non structurée ne produisent pas des codes séparables. Cela est dû au fait que les patrons sont bidimensionnels afin de garantir une robustesse dans toutes les directions (voir la section 6.5 du chapitre 6). Cependant, il est possible de réduire le nombre d'images projetées en limitant la recherche d'une correspondance à la ligne épipolaire. Ainsi, la taille de l'espace de recherche passe de $w * h$ à au plus $\sqrt{w^2 + h^2}$, la dimension de la diagonale d'une image de projecteur de dimension $w \times h$. Dans le prochain chapitre, nous donnons les détails techniques d'implantation de notre méthode basée sur la lumière non structurée en utilisant la contrainte épipolaire, qui permet en pratique de réduire le nombre de motifs utilisés d'à peu près la moitié.

Il est important de noter qu'une méthode ne devrait pas fonctionner uniquement si la géométrie épipolaire est connue, car celle-ci n'est pas toujours facile à estimer (par exemple si la caméra n'a pas un seul centre optique, ou si les objectifs utilisés sont catadioptriques). Dans ce cas, la méthode devrait quand même trouver des correspondances, quitte à projeter davantage de motifs.

Nous verrons aussi que chaque méthode possède un seuil d'images à partir duquel elle ne peut plus fonctionner correctement. Ce seuil dépend souvent de la scène à reconstruire. Dans le prochain chapitre, nous verrons que certaines méthodes ont des avantages par rapport aux autres. Lorsque peu d'images sont utilisées, il est avantageux de mélanger les motifs de différentes méthodes afin de profiter de ces avantages. Ainsi, au chapitre 8, nous proposons des méthodes hybrides qui utilisent la géométrie épipolaire et combinent les motifs de lumière non structurée aux motifs à déphasage pour produire des reconstructions précises et très robustes avec peu d'images.

7.2 Calcul de correspondances optimales

Pour comparer les reconstructions de plusieurs méthodes, l'approche conventionnelle consiste à utiliser un objet dont les dimensions sont connues et de calculer une distance entre le modèle 3D reconstruit et l'objet de référence. Dans l'article que nous présentons au chapitre suivant, nous argumentons qu'une telle erreur contient plusieurs termes qui ne dépendent pas de la méthode de correspondance active. En particulier, elle encapsule l'erreur de calibrage (due à une mauvaise estimation des paramètres de modélisation géométrique de la caméra et du projecteur). De plus, si l'objet de référence a été obtenu par une méthode externe (e.g. une numérisation laser), cette erreur comprend aussi une erreur d'alignement entre le modèle reconstruit et la référence. Ces erreurs ne sont pas indépendantes. Par exemple, l'algorithme qui aligne les deux modèles peut compenser des erreurs dans l'estimation de la distorsion

radiale en changeant l'échelle d'un modèle...

En fait, deux méthodes peuvent voir leur reconstruction alignée de manière différente, même si les mêmes paramètres de calibrage sont utilisés. Dans les articles précédents, nous avons utilisé à tour de rôle les résultats de notre méthode avec un très grand nombre d'images projetées, et un modèle 3D acquis à l'aide d'une méthode ultra précise, pour déterminer la précision de nos reconstructions. Dans le prochain chapitre, nous modifions une méthode connue pour caractériser les interactions entre le projecteur et la caméra : *la matrice d'illumination*, afin de calculer une correspondance de référence qui ne dépend d'aucun paramètre de calibrage. Ainsi les comparaisons entre les différentes méthodes et cette référence sont moins biaisées. Nous présentons brièvement les bases de cette méthode dans le prochain paragraphe.

Matrice d'illumination

Afin de mesurer les interactions complexes ayant lieu lors de la projection d'un motif lumineux dans la scène, la *matrice d'illumination* a été proposée. Cette matrice $\mathcal{M}_{(W^c H^c \times W^p H^p)}$ possède autant de lignes qu'il y a de pixels dans la caméra, et autant de colonnes qu'il y a de pixels dans le projecteur. Chaque colonne i de la matrice correspond à l'image qu'une caméra mesurerait si le i^{e} pixel du projecteur était allumé (voir figure 7.2). Ainsi, pour obtenir l'image qu'une caméra mesurerait si le projecteur allumait plusieurs pixels simultanément, il suffit de multiplier la matrice par un vecteur colonne $\boldsymbol{v}_p$ représentant l'image du projecteur : $\boldsymbol{v}_c = \mathcal{M}\boldsymbol{v}_p$. On obtient un vecteur colonne $\boldsymbol{v}_c$ qui représente l'image synthétique telle qu'elle serait observée par une caméra si l'image était projetée.

Cette méthode permet de mesurer toutes les propriétés de la scène, à la fois géométrique et photométrique. Ainsi, cela permet de modéliser les illuminations indirectes dues aux interréflexions, à la dispersion sous-surface, etc. Pour la mesurer, il est nécessaire d'allumer à tour de rôle chaque pixel du projecteur, et d'en faire une acquisition afin de remplir chaque ligne de $\mathcal{M}$. L'acquisition de cette matrice

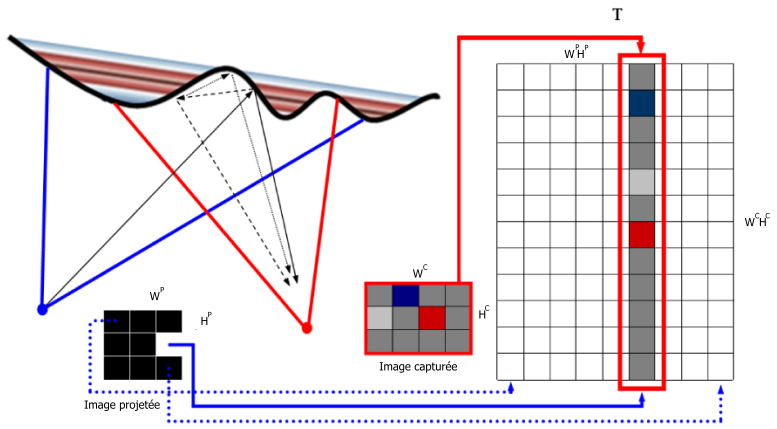


FIGURE 7.2: La matrice d'illumination $\mathcal{M}$ possède autant de lignes qu'il y a de pixels de caméra, et autant de colonnes qu'il y a de pixels de projecteur. Chaque colonne de $\mathcal{M}$ correspond à l'image, sous forme d'un seul vecteur, capturée lorsqu'un seul pixel de projecteur est allumé. Tirée de [7] (traduction libre).

prend beaucoup de temps et requiert un nombre important d'images, bien que des méthodes pour diminuer le nombre d'images requis existent[70, 63]. La matrice possède énormément d'entrées nulles et un stockage sous forme de matrice clairsemée est plus avantageux.

Dans la comparaison que nous proposons au chapitre suivant, nous avons utilisé ce principe pour capturer la correspondance géométrique entre la caméra et le projecteur. Comme nous l'avons mentionné auparavant, la relation géométrique entre la caméra et le projecteur, ainsi que la géométrie de la scène est encapsulée à travers cette matrice. En particulier, la correspondance *entière* entre le projecteur et la caméra peut facilement être trouvée en cherchant le pixel dont l'intensité est maximale dans l'image de la caméra lorsque chaque pixel de projecteur est allumé. La seule hypothèse que nous posons pour établir ce résultat est que l'intensité issue de l'illumination directe est toujours plus forte que celles issues des illuminations indirectes, lorsqu'un seul pixel de projecteur est allumé. Dans nos travaux futurs, nous tenterons

de montrer que cette hypothèse peut en fait être considérée comme une définition de ce qu'est l'illumination directe.

La correspondance entre la caméra et le projecteur est un peu plus difficile, car il faut dans ce cas chercher le pixel de projecteur tel que l'intensité d'un pixel de caméra est maximale. Donc, après avoir illuminé chaque pixel de projecteur un à un, celui qui a éclairé directement un pixel de caméra avec la plus forte intensité est choisi comme correspondance. Puisque chaque correspondance est trouvée indépendamment de celle du pixel voisin, et qu'elle est invariante à l'illumination indirecte puisqu'un seul pixel est allumé pour chaque capture, nous considérons qu'elle fournit une correspondance "optimale", au sens qu'aucune méthode de correspondance entière ne pourrait faire mieux. Il serait possible d'obtenir une correspondance sous-pixel en faisant la supposition supplémentaire de variation linéaire de l'intensité dans un voisinage. Dans nos travaux futurs, nous prévoyions plutôt de réduire les résolutions des images de la caméra et du projecteur artificiellement, lors de la comparaison avec la correspondance de référence. Dans ce cas, la référence deviendrait automatiquement plus haute résolution et permettrait de servir de référence pour une méthode sous-pixel.

Chapitre 8

A COMPARISON OF CODED LIGHT METHODS FOR PRECISE AND ROBUST ACTIVE RECONSTRUCTION (ARTICLE)

Ce chapitre présente l'article[43] en préparation pour publication tel que l'indique la référence bibliographique :

> N. Martin et S. Roy, A comparison of coded light methods for precise and robust active reconstruction, IEEE Transactions on Pattern Analysis and Machine Intelligence.2014, IEEE, 2014. Manuscript submitted for publication.

Cet article présente une comparaison des méthodes temporelles de reconstruction active. Une méthode de création d'une carte de correspondance de référence utilisée pour évaluer les performances de chaque méthode est présentée. Finalement, des méthodes hybrides comblant les motifs de lumière non structurée ainsi que les méthodes à déphasage sont proposées.

Cet article est en préparation pour une soumission à un journal scientifique. Une méthode pour capturer une carte de correspondance sous-pixels est en préparation et permettrait de compléter la section des expériences en ajoutant davantage de résultats quantitatifs.

L'article est présenté sous sa forme originale.

Abstract

3D reconstruction of an object using affordable hardware has been a challenge of growing interest for decades. Active light reconstruction provides a fast and accurate solution to this problem. In the past, several reviews of active light methods were

proposed, but in a very generic context. The major contribution of this paper is to assess the performance of dense subpixel active light reconstruction algorithms. Only temporal active light methods will be investigated in this work, since our main concern is accuracy and precision, not speed. Another aspect of our work is to compare reconstructions obtained in difficult setups, focusing on robustness to indirect illumination, scene discontinuities and hard-to-scan materials. We present a slow but simple method that captures “optimal" correspondences. It is used as a reference for our comparative evaluation of all the methods. Finally, when applicable, we evaluate how some methods behave as the number of patterns is reduced. We also present hybrid methods which combine the strength of other methods and performs better than current state of the art with the same amount of images.

8.1 Introduction

Active light reconstruction is the process of producing a 3D reconstruction of an object using correspondences between captured images of the object illuminated by several projected patterns. In general, the correspondence map between both images is what is computed by the method, while the 3D reconstruction is achieved through simple triangulation given both devices intrinsic parameters and relative pose[29].

In the past, several methods were proposed to recover a camera-projector correspondence map. In the classification of [57], each method is categorized as either a temporal or spatial multiplexed method. Every method that makes use of spatial information is bound to fail at discontinuities[51]. Most spatial methods only produce sparse 3D points at locations where the neighborhood information is reliable, thus requiring the remaining points to be interpolated with some heuristics. Some methods, called *grating*[57] only use the information in one pixel (and could be classified as using a neighborhood of size zero). They are sensitive to noise, since the disambiguation between two neighboring pixels rely on very few images[58]. They

also generally require the object to be weakly textured.

In general, spatial methods refer to methods that aim at producing a reconstruction using only one image for real-time purpose. While this approach is needed for reconstruction of dynamic scenes [1], the *one-shot* constraint is orthogonal to the quality and robustness constraints of reconstructions which can be achieved using temporal methods. Thus, in this review, we are only interested in temporally multiplexed methods for active light reconstruction.

While this choice restrains our comparison to reconstruction of static scenes, we believe that only for this type of scenes, both subpixel quality and robustness can be achieved. At the same time, it enables a fair comparison between methods, independently of the number of patterns required, since the object can remain static during the acquisition. Finally, temporal methods do not generally impose any constraint on the content of the scene (such as material, color, texture or depth range), and are therefore the most general class of methods applicable.

In Sec. 8.2, we review the previous works in temporal multiplexed methods for active light reconstruction, and present the methods which will be compared in the experiments. In Sec. 8.3, we present the camera-projector setup we use in the experiments and discuss the implementation details of the methods we chose to review. In Sec. 8.4, we present the methodology we use to evaluate the results of each methods and in Sec. 8.5, we compare the correspondences obtained for a scene and present both qualitative and quantitative results. Finally, in Sec. 8.6, we conclude and propose future work.

8.2 Previous work and state of the art

Here we review the most important work in active light reconstruction using multiple projected patterns. Temporal methods are most of the time called struc-

1. Scenes containing moving objects or a moving camera

tured light methods, that is methods for which the position of a projector pixel is encoded trough the structured temporal sequence of projected patterns. One class of structured light methods uses discrete coding[57]. The original method uses binary codes[53], later improved as Gray codes[33], is probably the quintessential example of discrete structured light. Each projector pixel is encoded using Gray codes, and each projected *black and white* pattern only contains the i^{th} bit. Several variations on these codes were proposed to alleviate the problem of accurately recovering the lowest bits from aliased pixels of the captured image[26]. Recently a very simple extension of these patterns enables subpixel reconstruction by exploiting the projector blur affecting the sequence of captured images[67]. In an effort to reduce the number of projected patterns, some methods [11] use *n-ary* color patterns, at the cost of requiring a careful photometric calibration. It was however shown that patterns based on binary codes are not robust, mainly against indirect illumination[15, 27]. The images corresponding to the highest bits use very low frequency and make the decoded bits less reliable because of the indirect lighting induced by these patterns [50]. The method of [27] uses several sets of high frequency xor-ed versions of Gray codes patterns to achieve a robust, yet non-subpixel, correspondence.

Another class of structured light method uses sinusoidal patterns, most of the time referred to as *phase-shifting*. The simplest method uses only one-frequency sinusoidal functions to encode projector positions[64]. At least three shifted versions of each sinusoidal pattern is needed to recover the phase, and thus the position of the projector pixel. There has been some work to embed those three shifts inside a single color image[75, 79]. However, the general consensus is that for added precision and robustness, more shifts are needed, especially if projector non-linearities are not corrected prior to projection[38]. Phase unwrapping is the most important problem with those methods, since recovered phases are not unique due to the periodic nature of the patterns. Some methods deal with this problem by requiring that the scene be smooth or depth bounded[79]. Several methods were introduced to compute

absolute phases based on spatial or temporal unwrapping[31]. Spatial unwrapping is very sensitive to noise, shadows or discontinuities[57]. On the other hand, temporal unwrapping requires the projection of several frequencies, from the lowest to the highest possible. It is therefore very sensitive to indirect illumination[50]. Micro phase shift [28] is a method that uses several high frequency sinusoids to disambiguate the phases while achieving robustness to indirect illumination. Another idea was proposed in [13], where the patterns were modulated by high-frequency sinusoids at the cost of a higher number of projected patterns.

Unstructured light patterns were used in [35] and formally introduced in [15]. These patterns do not directly encode the projector pixel position throughout the sequence of projected patterns. Instead, the temporal sequence of captured images are used to generate a codebook of intensity values, and the correspondences are found by formulating the problem as a high-dimensionality nearest neighboor search[15]. The specific patterns used are band-pass white noise patterns, designed to homogenize the amount of indirect illumination generated by the projection. Contrary to most temporal methods, the patterns are high frequency in both direction of the image, and the method outperforms the others in terms of robustness. The number of required patterns is however, much more than other state of the arts method, and the reconstruction is not subpixel. It was later improved by achieving subpixel correspondences through the use of quadratic coding, while requiring less patterns[42].

In the next section, we present the setup we used to compare the methods, and propose a simple automatic method to calibrate the projector-camera system. We also discuss the implementation details of each methods we use for our comparative evaluation.

8.3 Experimental setup

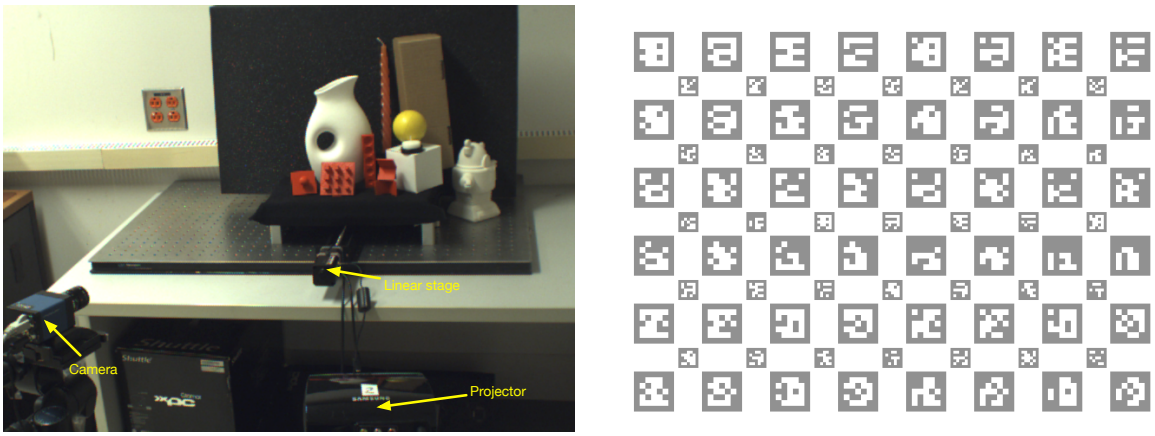
Camera-projector calibration, while not strictly required to produce camera-projector correspondences, is often performed to enable the subsequent 3D mesh generation. However, when it is available, it can also be embedded inside the method, thereby reducing by half the number of projected images. The following section provides a simple automatic calibration method that recovers both intrinsic parameters and relative camera-projector geometry using a linear translation stage.

8.3.1 Calibration

Several methods exist to calibrate a camera-projector system [78, 18, 48]. While any of them could have been used, none of them was fully automatic, requiring some kind of user input to perform accurately. We devise a simple automatic calibration method that only requires the use of a printed pattern moved automatically by a linear translation stage.

The idea is to have the camera detect some interest points and find their images in the projector, by the use of any coded light method. The pattern we used is a two-level hierarchic grid of simple fiducial markers as depicted in Fig. 8.1-a. Typically, a checkerboard is used to calibrate a camera, however we found that fiducial markers were better detected when the board was far from the camera, since the detection process does not fail even if some markers are not found. Compared to a checkerboard, there is also no orientation ambiguity, since each marker is oriented.

The pattern is mounted on a board which is moved on a translation stage at fixed positions, as in Fig. 8.1-b. At each step, the tags are detected and a correspondence map is computed between the camera and the projector using e.g. a phase-shifting method with several frequencies[81]. Once the correspondence map is found, each marker position can be transferred in the projector. Since the depth of the markers are easily derived from the position of the board at each step, the camera and projector



(a) (b)

FIGURE 8.1: (a) The camera-projector setup we have used in the experiment and the translation stage moving the calibration pattern. (b) The calibration pattern is made of two grid of fiducial markers of different size.

can be independently calibrated using a robust RANSAC DLT[29]. Finally, a bundle adjustment is carried out to recover the radial distortion parameters of the camera.

In the following section, we briefly review the implementation details of the methods we evaluate in Sec. 8.5.

8.3.2 Implementation details of the compared methods

In the Sec. 8.5, we compare several state of the arts active light reconstruction methods, as well as standard methods such as Gray codes, and simple phase-shifting.

Choice of methods

In our experiments, we compare state of the art methods : micro phase shifting (MPS)[28], unstructured quadratic patterns (UQ) and its subpixel version (UQS)[42]. We evaluate the modulated phase shifting (MODPS)[13] which was originally devised to be robust to indirect lighting. We also compare to standard structured light me-

thods : Gray codes (GC)[33] and several frequencies phase shifting (PS)[81]. Finally, we implemented the subpixel extension of Gray codes, called line shifting (LS)[67].

This choice of methods provides a pool of methods with various characteristics : some of these methods are robust to indirect lighting, some are subpixel, some perform well with few images. One of the aspect we focus on in the experiments is the behavior of methods as the number of patterns is reduced. Gray codes and its subpixel variant use a fixed number of images. On the other hand, all the methods based on unstructured light and phase shifting can accommodate more and more images. To further reduce the number of images, some of these methods can use 1D only patterns. This is not the case of unstructured light patterns. In the next subsection, we show how to implement UQ and UQS using less patterns, by exploiting the epipolar geometry.

Reducing the number of images : 1D codes

When the epipolar geometry is known, the correspondence of a point is restricted to its corresponding epipolar. In this case, the coded light method needs only to encode one dimension roughly orthogonal to the baseline joining the camera centers. This reduces the required number of projected patterns of most methods by half. To compute the correspondences, one can rectify the images using the epipolar geometry, and then match the images in rectified space. Instead, we prefer to solve the missing dimension by finding the intersection between the epipolar line and the decoded dimension. This has the advantage of not having to rectify the images, and to obtain a correspondence map which has the same resolution as the camera, rather than some arbitrary resolution corresponding to the plane of projection chosen during rectification[20].

When the patterns are bidimensional, as is the case for UQ, one-dimensional encoding cannot directly be achieved. Since we cannot extract 1D only information, we propose to embed the knowledge of the epipolar geometry directly in the matching

algorithm[15]. The idea is to restrict the matching algorithm to search for a correspondance in the corresponding epipolar lines, rather than across the whole image. Each pixel in the camera and the projector is assigned a discrete label corresponding to its angular position in the epipolar pencil[55]. Then the matching algorithm is applied between pixels of similar labels. In our implementation, pixels having a distance of a most 3 pixels are considered. The LSH[4] matching algorithm is thus performed once for every epipolar lines, instead of once for the whole image, but with far less pixels in the table of the camera and the projector. The matching heuristics proposed in[15] are also applied after the whole image has been matched using this modified algorithm.

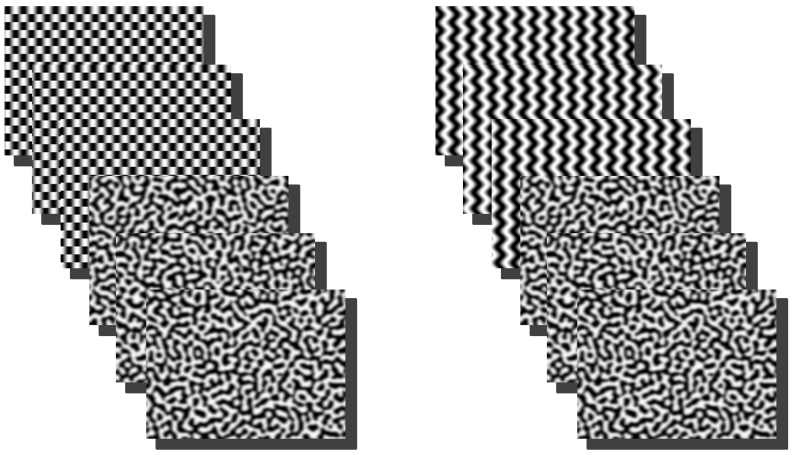
With this modified matching algorithm, it is possible to use less patterns for the UQ method. In fact, a little more than half of what is required for 2D matching is used in practice. In theory, half of them should be enough as for one-dimensional methods, but the local coherence of UQ patterns make the disambiguation of neighboring pixels a more challenging task, requiring more patterns than structured light methods. The UQS method however, requires many more patterns to produce subpixel correspondences, since the quadratic code generation[42] is more beneficial when the number of patterns is high. In fact, we found that UQS perform very well when more than 30 patterns were used, indicating that UQS is not optimal for subpixel correspondence when a limited budget of images is available. Instead, we propose to mix UQS and PS based patterns to produce accurate and robust subpixel correspondences using few images.

Hybrid methods

Phase shifting is a very efficient method to recover dense subpixel correspondences. It suffers from two majors drawbacks : it needs some kind of phase unwrapping to work on scene featuring a wide range of depths, and it uses one-dimensional patterns which, when the frequency is high enough, provide some robustness but

only in one direction. Phase unwrapping can be implemented through several methods, and the solution used by MPS[28] is quite efficient. However, it suffers from the same shortcoming, as it uses one dimensional high frequency patterns. UQ[42] can however be used to compute absolute references to unwrap phases which are not affected by indirect lighting. Using the 1D matching algorithm we presented in the previous subsection, and a median filtering to account for noise, as low as 10 images are needed to provide a truly robust phase unwrapping. Note that the median filter is used only to correct spurious correspondences due to the use of few images. Since our goal is to use UQ only for unwrapping purposes, errors inside the period used by a phase shifting method are not important, since the subpixel position will be recovered independently.

A problem remains in the form of PS patterns still being one-dimensional. One solution could be to modulate each patterns[13]. However, this would require too many patterns since each pattern is itself modulated by several others. Another solution is to modulate the sinus by only one modulating signal which varies as a function of the orthogonal direction. An example of such a pattern is given in[50] which uses the function $\sin(x + \sin y)$ as a modulating signal. Another example is to use a phase offset which varies as a function of the orthogonal phase $\sin(x + \pi * \mathrm{sgn}(\sin y)/2 + 0.5)$. Figure 8.2 gives an example of the patterns we use in our so-called hybrid methods which combine UQ and modulated PS patterns for maximal robustness and subpixel precision. Both of these patterns are high frequency in both dimensions. The phase recovery uses the absolute phase in the y direction computed using UQ patterns, to compute the wrapped but precise phase in the x direction. This phase is also unwrapped using absolute reference given by the unwrapping using UQ.



(a) (b)

FIGURE 8.2: The hybrid methods UQS-PS-A (a) and UQS-PS-B (b) use a mix of UQ patterns for unwrapping purpose, and modulated PS patterns to robustly estimate the subpixel phase.

8.4 Evaluation methodology

In this section, we motivate our choice of comparing only the correspondences rather than the 3D reconstructions. In order to provide a reference correspondence map which is not biased toward any method, we expose the method we used to acquire a groundtruth for the scene we used in Sec. 8.5. We also present the scene we used which features several 3D printed objects designed to test the performance of each method.

8.4.1 Comparison metric

To assess the performance of a reconstruction method, it is standard practice to provide 3D metric errors between a reconstructed object and its groundtruth version. However, a metric distance alone does not really express the quality of a

reconstruction. For example, reconstructing a sphere of diameter $5cm$ with a precision of $1mm$ is not really a challenge when the camera-projector system is itself located at a distance of $10cm$ of the sphere. At a distance of $5m$, this could be considered a very precise reconstruction. Hence, RMS errors given without a detailed description of the acquisition setup is not very useful.

Another issue arising when providing metric errors is that the 3D error encapsulates other errors which are not caused by the reconstruction method. For example, it inherently contains the geometric calibration error which can cause a lot of distorsion when the model is triangulated from correspondences (if the camera radial distorsion is not well estimated for example). If the groundtruth was acquired with some other, more precise method, another error can be introduced when trying to align both the reconstructed mesh and the groundtruth. In this case, the situation is worse, since the alignment error will try to compensate for calibration errors.

We propose to compare the correspondence maps themselves. Since the goal of an active light reconstruction is to compute the correspondence map, it makes sense to only compare those. This is a metric that has also been used in the past, however the groundtruth is not easily acquired, since it should be computed with the same setup, while most groundtruth methods require separate hardware. To deal with this problem, we modify the well known light transport matrix (LTM)[7] acquisition to only recover geometric correspondences, leaving out photometric and radiometric information.

8.4.2 Acquisition of a groundtruth correspondence map

The full light transport matrix express the relationship between each projector and camera pixel. It is mostly used for relighting and radiometric compensations applications[7]. Its acquisition requires the projection of as many patterns as the number of projector pixels, although some techniques can be used to speed up the process[70, 63]. For our application, storing the full matrix is not required, since we

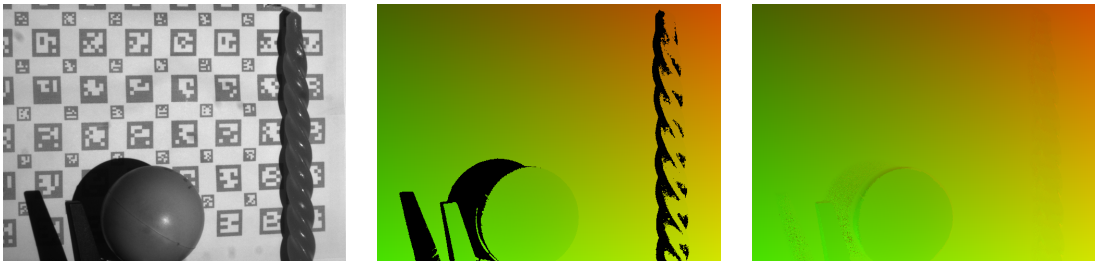
are not interested in synthesizing camera images of a projected pattern, nor solving for the optimal projection with respect to the camera. However, we can use the same process, keeping only the geometric information linking each projector pixel to camera pixels.

When turning on a single pixel of the projector, the captured intensities can take any form due to indirect lighting, surface material, occlusions and shadows... For each projector pixel, it's easy to isolate the camera pixel which has the highest intensity. This would be the corresponding pixel to the projector pixel that is turned on. Instead, we are interested in extracting the camera-projector correspondence map.

The only assumption we use to acquire the groundtruth is that for any point in the scene, its observed intensity is at its highest when it is illuminated directly. In other words, even if the pixel is sometimes illuminated by indirect lighting, the intensities measured in these cases can never be higher than the one measured when directly lit by a projector pixel. This assumption could be violated if we allow several projector pixel to be on, since the sum of their contributions to indirect lighting could be higher than the direct illumination. Thus, to simplify the process, we don't use any speed up techniques to acquire the LTM, and simply turn on each projector pixel one y one. To find the corresponding projector pixel of a camera pixel, we simply keep a map of the projector position which caused the highest intensity so far, updating this position as each projector pixel is turn on. The figure 8.3 shows a test scene, for which we computed a groundtruth as well as a correspondence map computed using the PS method.

8.4.3 Dataset

The scene we used for experiments is composed of several objects of varying shapes and materials, and is shown in Fig. 8.4-a. It is composed of soft discontinuities on the boundaries of the sphere made of rubber (A) and of plastic (B), hard edges

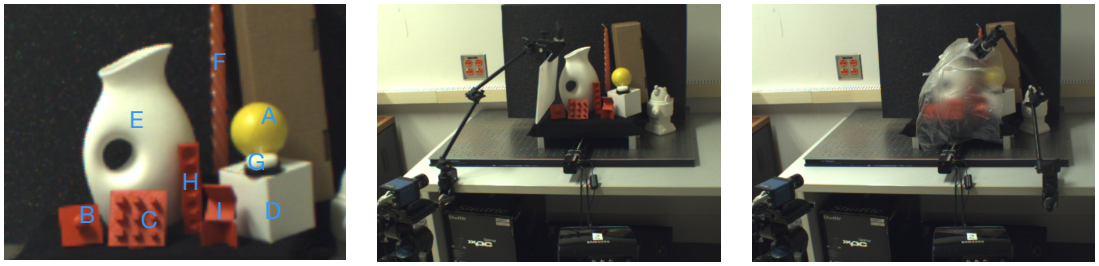


(a) (b) (c)

FIGURE 8.3: Correspondence map acquired with our groundtruth method on a test scene which contains a low albedo object, a soft edge sphere with specular highlight and a wax candle. (a) Reference image of the scene (b) Groundtruth acquired with our modified LTM method and (c) Correspondence map computed with PS.

and sharp discontinuities on the boundaries of the plastic towers (C) and the box (D), lambertian objects like the vase (E), subsurface scattering like the wax candle (F) and the less diffuse candle (G), convex cavities (H) and oriented corners (I). It is also shown in Fig. 8.4-b and Fig. 8.4-c with added occluders.

We scanned a groundtruth of the scene once using the method of Sect. 8.4.2, and then we captured the scene for each methods with and without the occluders. For example, we added a plane to the scene on the left, that is reflecting the light from the projector on the scene, creating some interreflections on the vase and the plate with a sphere attached to it. This occluder is shown in Fig. 8.4-b. Another occluder we added is simply a transparent plastic bag, which absorbs some of the light that illuminates it (see Fig. 8.4-c), but also act as a strong diffuser, thereby inducing an effect similar to subsurface scattering.. For both occluders, the idea is to test the robustness of methods when the scene is artificially degraded. The advantage is that the groundtruth is the same whether the occluder is present or not, so it is easy to evaluate the robustness of the methods by looking at the variation between the reconstructions with and without the occluder, and the same groundtruth corres-



(a) (b) (c)

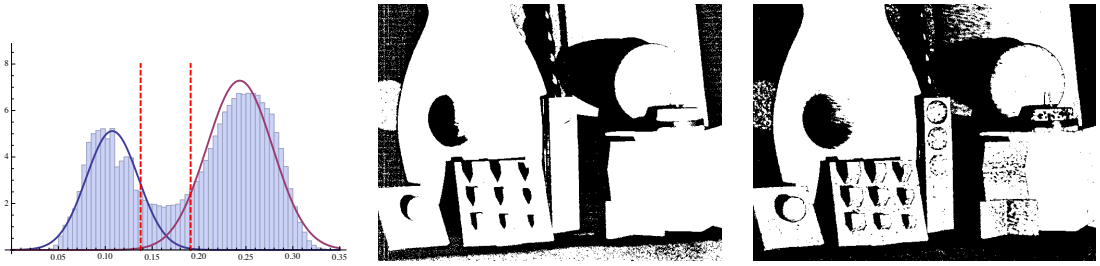
FIGURE 8.4: The scene we used in our experiments contains several objects and materials to evaluate the robustness of each methods. (a) Close-up of the scene labelling each objects (see text), (b) The scene with a plane occluder creating indirect lighting directed towards some objects. (c) The scene with a transparent plastic bag covering the objects.

pondence can be used. In the next section, we present the results of our comparative evaluation using this scene.

8.4.4 Masking

It's worth noting that the assumption we used to compute the groundtruth can fail for some pixels. This is because correspondences will be found for every camera pixel that was sufficiently illuminated during the whole LTM acquisition. When the pixel is directly illuminated, the correspondence found will be the correct one. However, when a camera pixel is in an area that is indirectly illuminated (e.g. a shadow), the correspondence found will be erroneous and should be discarded.

The identification of those pixels is not a trivial task. A common practice is to create a mask by segmenting the pixels based on some criteria, for example the standard deviation of the captured intensities[15]. This works well in practice if the projected intensities are well distributed between black and white. For example, this mask can be computed by projecting several pure white noise patterns. Fig. 8.5



(a) (b) (c)

FIGURE 8.5: Mask used to separate valid correspondences from invalid ones due to indirect lighting. (a) Histogram of normalized standard deviations of captured intensities for a sequence of white noise patterns. The distribution should be bimodal and represent pixel directly and indirectly illuminated. (b) Mask computed using a threshold of $t = 0.14$. (c) Mask computed using a threshold of $t = 0.19$. Notice that no value of t can perfectly separate indirectly illuminated pixels from pixels with low albedos.

shows the histogram of normalized standard deviations of captured intensities for a sequence of white noise patterns. Notice that the distribution can be approximated by a mixture of two gaussian distributions that represent respectively the pixels which are directly and indirectly illuminated. The mask is then computed by finding an appropriate threshold that separates pixels that are indirectly illuminated from pixels for which the correspondence is valid. As can be seen in Fig. 8.5, a threshold that fully separates both kind of pixels does not exist since it is impossible to differentiate a pixel indirectly illuminated from a directly lighted pixel with a very low albedo. If the albedo was available, then a better mask could be computed by combining this information with the standard deviation of intensities. In practice, a trade-off between correspondence density and accuracy must be made and for our experiments we chose the mean value of both distribution tails as a threshold ($t = 0.175$). In Sec. 8.5, we use the mask presented in Fig. 8.5 to compare the results

of each method with the groundtruth. If a pixel has a correspondence outside of this mask, it should be considered as an error, since it can not be reliably estimated by the groundtruth method. It is important to note that few methods compute a masked correspondence map, and actually leave the decision of which correspondence is valid to some other process. Most of the time, this process is a human manually discarding false correspondences by segmenting depths of the reconstructed model. This is far from convenient when using an automated algorithm to compute the 3D meshes. The method we use to compute the mask is only valid for projected patterns with well distributed intensities. Because of this, masks computed with low frequency patterns will generally not be as accurate, and will either discards valid correspondences or accepts too many false positives.

8.5 Experiments

The scene presented in Fig. 8.4-a was first scanned with our groundtruth method. We then added a plane occluder in the scene in order to produce indirect lighting bouncing on the left side of the vase, as seen in Fig. 8.4-b. The correspondence maps computed with each methods are presented in Fig. 8.6. We then evaluated the difference between the correspondences computed with each method and our groundtruth reference for some of the parts labeled. The results are presented in Fig. 8.7 and Fig. 8.8. Note that since the groundtruth is not acquired with subpixel precision, we count a correspondence as an error if its absolute difference with the groundtruth is more than 1 pixel. Because of this, correspondences obtained with GC (which is not subpixel) are not visually worse than correspondences computed with other methods. In order to better assess this difference, the groundtruth should be acquired with subpixel precision, or the correspondences should be synthetically downsampled as mentioned in Sec. 8.6.

First, notice that the differences between all the methods tend to be small, es-

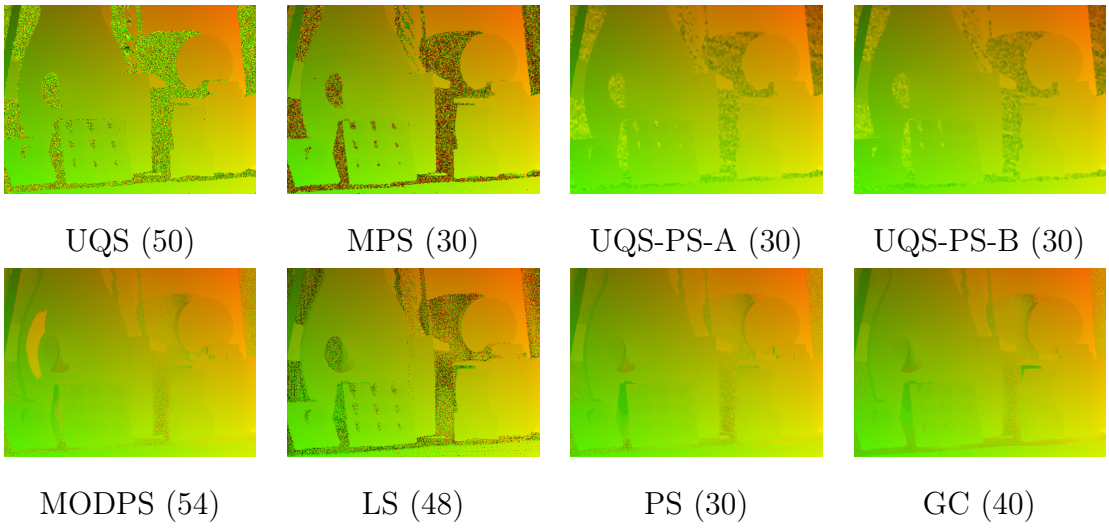
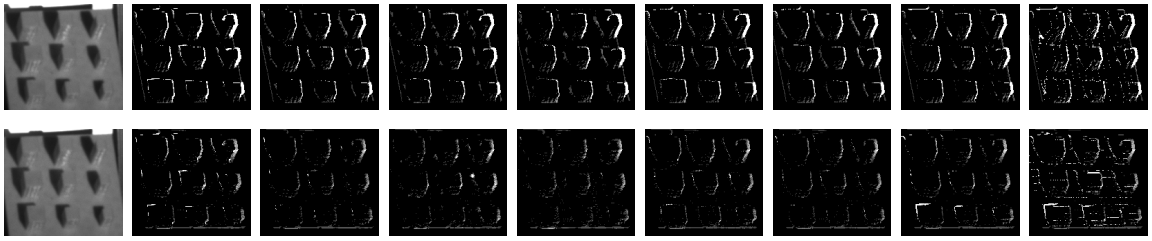
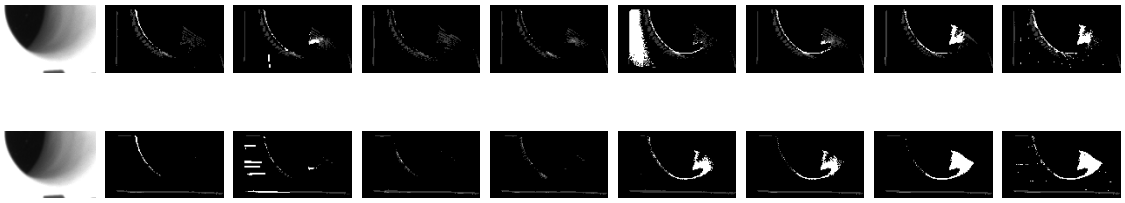


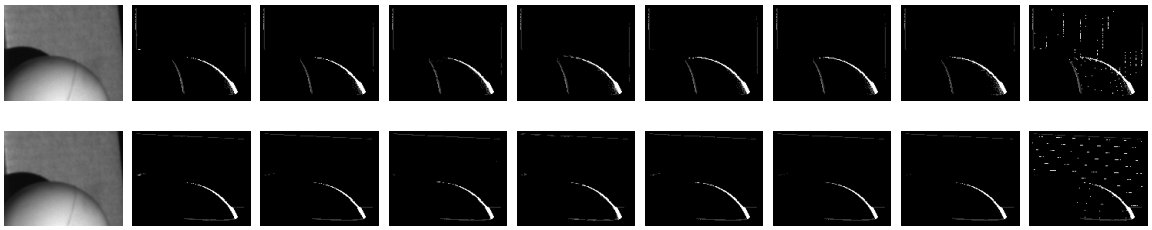
FIGURE 8.6: Correspondence maps computed for each method on the scene presented in Fig. 8.4. The number next to the method name is the number of patterns used by the method. The top row correspond to methods which are robust against indirect illumination : UQS, MPS and the hybrid methods. The bottom row features moderately robust methods : MODPS and LS (due to unwrapping using low frequency) and methods which are not robust : PS and GC. The correspondence maps are not filtered and the mask found with our groundtruth method will be applied before the comparisons.



(a) Sharp plastic towers labeled C in Fig. 8.4-a.

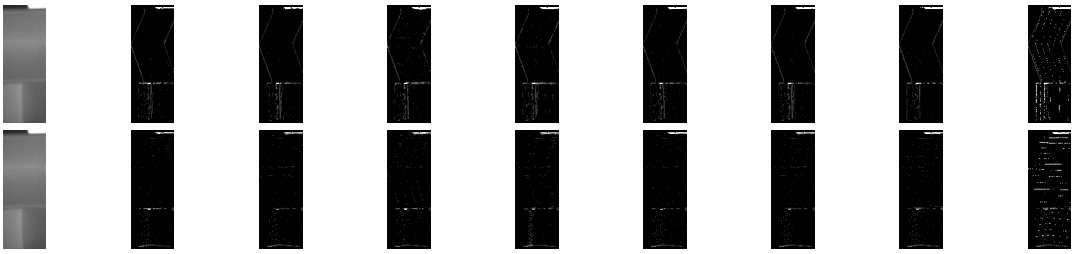


(b) Lambertian vase labeled E in Fig. 8.4-a.



(c) Soft edge rubber ball labeled A in Fig. 8.4-a.

FIGURE 8.7: Absolute difference for cropped regions of interest labeled C (a), E (b) and A (c) in Fig. 8.4-a. A correspondence is considered erroneous if the absolute difference with the groundtruth is at least 1. The top (resp. bottom) row corresponds to absolute difference in the X (resp. Y) direction. The methods compared are from left to right : UQS, MPS, UQS-PS-A, UQS-PS-B, MODPS, PS, GC, LS.



(a) Oriented corners labeled I in Fig. 8.4-a.

(b) Lambertian vase labeled E in Fig. 8.4-a.

FIGURE 8.8: Absolute difference for cropped regions of interested labeled I (a), E (b) in Fig. 8.4-a. A correspondence is considered erroneous if the absolute difference with the groundtruth is at least 1. The top (resp. bottom) row corresponds to absolute difference in the X (resp. Y) direction. The methods compared are from left to right : UQS, MPS, UQS-PS-A, UQS-PS-B, MODPS, PS, GC, LS.

pecially in areas not affected by indirect illumination (see Fig. 8.7-(a) and (c), and Fig. 8.8-(b). One of the reasons is that differences between correspondences are only visible at a subpixel level which can not be evaluated with the groundtruth reference. However, for parts which were affected by the occluder, the correspondences for robust methods are much less prone to errors, as expected. UQS and the hybrid methods are the least affected methods in Fig. 8.7-b. This is due to the fact that they use patterns which are high frequency in both directions. MPS compute correspondences in the hollow and slanted part of the vase which receives some indirect light. The unwrapping computed with MPS is also erroneous for some pixels (bottom row of Fig. 8.7-b and Fig. 8.8-b). Non robust methods compute false correspondences in every parts of the vase when it is affected by indirect lighting. Even if MODPS does use high frequency patterns to compute the subpixel phase, it still uses low frequency sinusoidal patterns to compute the unwrapping and as such computes wrong correspondences too. Notice that the occluder created an overflow in the camera intensities on the left part of the vase which seems to only affect MODPS (bottom row of Fig. 8.7-b). Finally LS makes some periodic mistakes due to the unwrapped positions computed using Gray codes, which are affected by indirect lighting, and combining them with subpixel shifts which are not, thus creating artefacts appearing approximately every 8 pixels (which is the space between two line stripes in the patterns used in LS). Using the same number of images as MPS and PS, the hybrid methods perform better in areas affected by indirect lighting.

We also present the correspondences found on the scene using the bag occluder in Fig. 8.9. Notice that robust methods (top row) actually reconstructs the bag, whereas non robust methods (bottom row) sometimes reconstruct the objects behind the bag when they are close enough. This is an artefact induced by diffused light bouncing on the objects behind the bag and then illuminating the bag from behind as indirect lighting.

Finally, we evaluate the evolution of the subpixel correspondences as the number

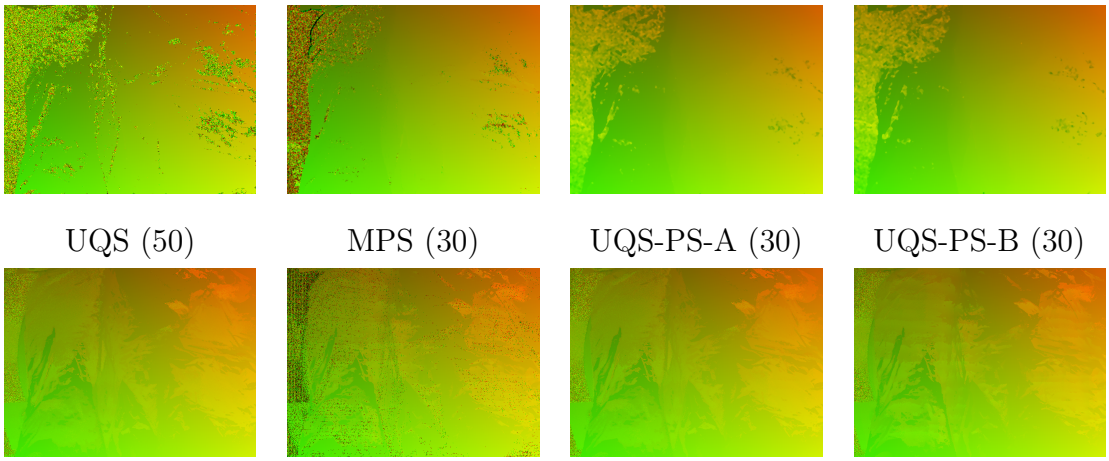


FIGURE 8.9: Correspondence maps computed for each method on the scene presented in Fig. 8.4-c. The number next to the method name is the number of patterns used by the method. The top row correspond to methods which are robust against indirect illumination : UQS, MPS and the hybrid methods. The bottom row features moderately robust methods : MODPS and LS (due to unwrapping using low frequency) and methods which are not robust : PS and GC. The correspondence maps are not filtered and the mask found with our groundtruth method will be applied before the comparisons.

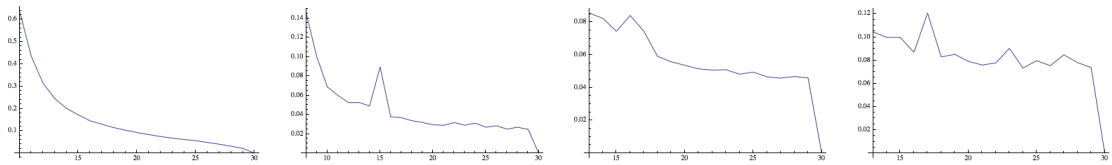
of projected patterns varies. Only robust methods were compared in this test. Since the groundtruth can not be used as a reference for subpixel comparison, we compare each method with itself. This comparison does not provide a way to compare each method with each other, however it enables us to assess the quantity/quality ratio. As more and more patterns are added, every method will improve its results until reaching a stable solution. If a method reaches this state early, it can perform well with very few patterns. If a method does not really improve this ratio as more and more patterns are added, then there is no point in using a higher number of patterns.

We compare each method [2] with itself, by computing the absolute differences between the correspondence map obtained with the maximum number of patterns (30). The results are presented in Fig. 8.10. It is clear from this figure that UQS needs a lot of patterns to produce good subpixel correspondences. It was shown that as more and more patterns are used, the correspondences are better and the method even becomes more robust to gamma and noise[42]. On the other hand, MPS and the hybrid methods produce very precise subpixel correspondences very quickly. As more and more patterns are used, each method produces better reconstructions, but some methods need more patterns than other achieve their best performance. As it was shown in the previous experiments that UQ is more robust to indirect illumination, but phase-shifting based methods produce better subpixel with fewer patterns, it makes sense that hybrid methods combine both of these patterns to produce very good correspondences with very few patterns.

8.6 Discussion and future work

We proposed an evaluation of the performance of current state of the art active light reconstruction methods both in terms of robustness and precision. We demonstrated that a simple modification of the LTM method can be used to acquire

2. we used the 1D version here to lower even more the number of images used



(a) UQS (b) MPS (c) UQS-PS-A (d) UQS-PS-B

FIGURE 8.10: Mean absolute differences of each method with itself for varying number of patterns. Horizontal axis : number of images, vertical axis : mean absolute differences in pixels between correspondences of the method with itself using 30 patterns.

a pixel precision groundtruth. This reference correspondence map can be used to compare the methods without introducing any bias related to the calibration or the alignment of 3D models to compute a 3D metric. Hybrid methods combining UQ patterns for unwrapping purposes and modulated phase shifting to compute precise subpixel phases were shown to outperform the other methods with an equivalent number of images in terms of robustness. In the future, acquiring subpixel groundtruth seems essential to evaluate the subtle differences between methods. Another more straightforward approach would be to artificially downsample the resolutions of both the camera and projector images. This would make the original groundtruth appear subpixel relative to the lower resolution images, allowing a better assessment of subpixel correspondences. Finally, we could evaluate the behavior of each method while using even more patterns. It would enable us to decide if some methods have a maximum precision level they can reach, from which adding more patterns truly do not improve the results.

CONCLUSION

Les travaux présentés dans cette thèse visaient non seulement à faire avancer l'état de l'art dans le domaine de la reconstruction 3D par de nouvelles méthodes de mise en correspondance, mais aussi à proposer un cadre rigoureux pour l'évaluation des performances de l'ensemble des méthodes.

À une époque où les imprimantes 3D foisonnent, deviennent économiquement rentables pour un particulier et où le besoin de tout stocker au format numérique est omniprésent, nous pensons que nos contributions ont une valeur qui n'est pas uniquement académique. Les scanneurs lasers produisent des reconstructions 3D de très bonne qualité, mais sont assez onéreux. Nous pensons que la mise au point d'un scanneur abordable, basé sur les motifs de lumière non structurée, serait tout à fait réalisable, en raison de la qualité et la robustesse des reconstructions que notre méthode produit à l'aide de matériel relativement bon marché.

D'un point de vue académique, les patrons de lumière non structurée ont été présentés suite à l'observation que les méthodes de lumière structurée existantes étaient déjà de très bonne qualité, mais peinaient sur le plan de la robustesse à l'illumination directe. Ce n'est pas un problème isolé et difficile à reproduire, comme nous l'avons montré dans chacune de nos expériences en reconstruisant des scènes composées d'objets de la vie de tous les jours. Pour des applications industrielles, les erreurs produites par les méthodes non robustes sont tout simplement inacceptables.

Pour atteindre les niveaux de précision des méthodes concurrentes tout en conservant la robustesse de la lumière non structurée, nous avons mis au point une méthode qui produit des reconstructions dont la précision est très élevée. En combinant robustesse et précision, les motifs de lumière non structurée basés sur les codes quadratiques constituent l'état de l'art en reconstruction active à l'aide de lumière codée.

Finalement, nous avons comparé notre méthode avec les meilleures méthodes de lumière codée afin d'isoler les forces et les faiblesses de chacune. Nous avons proposé une méthode permettant de construire une correspondance de référence qui permet une comparaison bidimensionnelle non biaisée qui pourrait servir à d'autres chercheurs souhaitant évaluer la qualité des reconstructions produites par leurs méthodes. Nous avons aussi proposé des méthodes hybrides qui permettent d'obtenir des reconstructions de très bonne qualité, tout en assurant une robustesse maximale aux défis majeurs des méthodes de lumière codée en utilisant aussi peu qu'une quinzaine d'images. En combinant robustesse, précision et nombre réduit d'images projetées, nous pensons que la lumière non structurée est une approche à privilégier pour la reconstruction 3D à l'aide d'un projecteur vidéo.

Plusieurs avenues de recherche restent cependant à explorer. En particulier, il serait très intéressant d'étudier la possibilité de varier localement les fréquences spatiales des motifs, afin d'offrir une robustesse accrue aux différents ratios de pixels caméra-projecteur liés à la profondeur des objets de la scène. Cela permettrait aussi d'améliorer les reconstructions des matériaux les plus difficiles, comme les objets à dispersion sous-surface qui nécessitent l'utilisation locale de fréquences plus basses. En développant notre méthode de correspondances "optimale", il nous est apparu évident que la mise en place d'un mode opératoire pour permettre aux autres chercheurs d'évaluer leur propre méthode sur les données d'une scène mise en place par d'autres groupes de chercheurs serait extrêmement bénéfique au domaine. Ce n'est pas un problème facile, mais la capture et la diffusion de la matrice d'illumination serait certainement une approche envisageable afin de permettre à n'importe qui de synthétiser les images dont sa méthode a besoin sans avoir à procéder à une capture réelle. La mise en place d'un banc de comparaisons a accéléré la maturité d'un très grand nombre de domaines en vision[60, 61, 6]. Nous espérons que les travaux de cette thèse contribueront de façon similaire à l'évolution de la reconstruction active.

RÉFÉRENCES

[1] http://vision3d.iro.umontreal.ca/en/projects/unstructured-light-scanning/.

[2] S. Agarwal et K. Mierle, *Ceres Solver : Tutorial & Reference*, Google Inc.

[3] A. Alexandr, *Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions*, *dans* IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, 2006, p. 459–468.

[4] A. Andoni et P. Indyk, *Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions*, Communications of the ACM, 51 (2008), p. 117–122.

[5] O. Arikan, D. A. Forsyth et J. F. O'Brien, *Fast and detailed approximate global illumination by irradiance decomposition*, *dans* ACM Transactions on Graphics (TOG), vol. 24, ACM, 2005, p. 1108–1114.

[6] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black et R. Szeliski, *A database and evaluation methodology for optical flow*, International Journal of Computer Vision, 92 (2011), p. 1–31.

[7] O. Bimber, D. Iwai, G. Wetzstein et A. Grundhöfer, *The visual computing of projector-camera systems*, *dans* ACM SIGGRAPH 2008 Classes, SIGGRAPH '08, New York, NY, USA, 2008, ACM, p. 84 :1–84 :25.

[8] K. L. Boyer et A. C. Kak, *Color-encoded structured light for rapid active ranging*, IEEE Transactions on Pattern Analysis and Machine Intelligence., PAMI-9 (1987), p. 14–28.

[9] R. N. Bracewell, *The fourier transform and its applications*, (1965).

[10] B. Carrihill et R. Hummel, *Experiments with the intensity ratio depth sensor*, Computer Vision, Graphics, and Image Processing, (1985).

[11] D. Caspi, N. Kiryati et J. Shamir, *Range imaging with adaptive color structured light*, IEEE Transactions on Pattern Analysis and Machine Intelligence., 20 (1998), p. 470–480.

[12] G. Chazan et N. Kiryati, *Pyramidal intensity-ratio depth sensor*, Department of Electrical Engineering, Technion Israel Institute of Technology, 1995.

[13] T. Chen, H.-P. Seidel et H. P. A. Lensch, *Modulated phase-shifting for 3d scanning*, *dans* IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2008, 2008.

[14] V. Couture, *Le cinéma omnistéréo ou l'art d'avoir des yeux tout le tour de la tête*, Thèse doctorat, Université de Montréal, Montréal, Québec, Décembre 2011.

[15] V. Couture, N. Martin et S. Roy, *Unstructured light scanning to overcome interreflections*, *dans* IEEE Computer Society International Conference on Computer Vision (ICCV)2011, IEEE, 2011, p. 1895–1902.

[16] V. Couture, N. Martin et S. Roy, *Unstructured light scanning robust to indirect illumination and depth discontinuities*, International Journal of Computer Vision, (2014), p. 1–18.

[17] J. Davis, D. Nehab, R. Ramamoorthi et S. Rusinkiewicz, *Spacetime stereo : A unifying framework for depth from triangulation*, IEEE Transactions on Pattern Analysis and Machine Intelligence., 27 (2005), p. 296–302.

[18] J. Draréni, S. Roy et P. Sturm, *Methods for geometrical video projector calibration*, Machine Vision and Applications., (2012).

[19] N. Durdle, J. Thayyoor et V. Raso, *An improved structured light technique for surface reconstruction of the human trunk*, *dans* IEEE Canadian Conference on Electrical and Computer Engineering. CCECE 1998, vol. 2, may. 1998, p. 874–877.

[20] A. Fusiello, E. Trucco et A. Verri, *A compact algorithm for rectification of stereo pairs*, Machine Vision and Applications., 12 (2000), p. 16–22.

[21] A. Gionis, P. Indyk et R. Motwani, *Similarity search in high dimensions via hashing*, *dans* VLDB '99 : Proceedings of the 25th International Conference on Very Large Data Bases, San Francisco, CA, USA, 1999, Morgan Kaufmann Publishers Inc., p. 518–529.

[22] L. Goddyn et P. Gvozdjak, *Binary gray codes with long bit runs*, Electronic Journal of Combinatorics, 10 (2003), p 27.

[23] S. J. Gortler, R. Grzeszczuk, R. Szeliski et M. F. Cohen, *The lumigraph*, *dans* Conference on Computer Graphics and Interactive Techniques. SIGGRAPH96, New York, NY, USA, 1996, ACM, p. 43–54.

[24] P. M. Griffin, L. S. Narasimhan et S. R. Yee, *Generation of uniquely encoded light patterns for range data acquisition*, Pattern recognition, 25 (1992), p. 609–616.

[25] J. Gu, T. Kobayashi, M. Gupta et S. K. Nayar, *Multiplexed Illumination for Scene Recovery in the Presence of Global Illumination*, *dans* IEEE Computer Society International Conference on Computer Vision (ICCV)2011, Nov 2011, p. 1–8.

[26] J. Gühring, *Dense 3-d surface acquisition by structured light using off-the-shelf components*, Videometrics and Optical Methods for 3D Shape Measurement, (2001).

[27] M. Gupta, A. Agrawal, A. Veeraraghavan et S. G. Narasimhan, *Structured light 3d scanning in the presence of global illumination*, *dans* IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2011, IEEE Computer Society, 2011, p. 713–720.

[28] M. Gupta et S. K. Nayar, *Micro Phase Shifting*, *dans* IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2012, Jun 2012, p. 1–8.

[29] R. Hartley et A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, ISBN : 0521540518, second éd., 2004.

[30] C. Hermans, Y. Francken, T. Cuypers et P. Bekaert, *Depth from sliding projections*, *dans* IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2009, 2009, p. 1865–1872.

[31] J. Huntley et H. Saldner, *Error-reduction methods for shape measurement by temporal phase unwrapping*, JOSA A, (1997).

[32] J. M. Huntley et H. O. Saldner, *Temporal phase-unwrapping algorithm for automated interferogram analysis*, Applied Optics, 32 (1993), p. 3047–3052.

[33] S. Inokuchi, K. Sato et F. Matsuda, *Range imaging system for 3-d object recognition*, *dans* International Conference on Pattern Recognition. ICPR 1984, 1984, p. 806–808.

[34] T. R. Judge et P. Bryanston-Cross, *A review of phase unwrapping techniques in fringe analysis*, Optics and Lasers in Engineering, 21 (1994), p. 199–239.

[35] A. Kushnir et N. Kiryati, *Shape from unstructured light*, *dans* 3DTV07, 2007, p. 1–4.

[36] P. Lavoie, D. Ionescu et E. Petriu, *A high precision 3d object reconstruction method using a color coded grid and nurbs*, *dans* Image Analysis and Processing, 1999. Proceedings. International Conference on, 1999, p. 370–375.

[37] M. Levoy et P. Hanrahan, *Light field rendering*, *dans* Conference on Computer Graphics and Interactive Techniques. SIGGRAPH96, New York, NY, USA, 1996, ACM, p. 31–42.

[38] K. Liu, Y. Wang, D. Lau, Q. Hao et L. Hassebrook, *Gamma model and its analysis for phase measuring profilometry*, JOSA A, (2010).

[39] K. Liu, Y. Wang, D. L. Lau, Q. Hao et L. G. Hassebrook, *Dual-frequency pattern scheme for high-speed 3-d shape measurement*, Optics express, 18 (2010), p. 5229–5244.

[40] K. Liu, Y. Wang, D. L. Lau, Q. Hao et L. G. Hassebrook, *Gamma model and its analysis for phase measuring profilometry*, Journal of the Optical Society of America A, 27 (2010), p. 553–562.

[41] Y. Ma, S. Soatto, J. Kosecka et S. S. Sastry, *An invitation to 3-D vision : From images to geometric models*, SpringerVerlag, Jan 2004.

[42] N. Martin, V. Couture et S. Roy, *Subpixel scanning invariant to indirect lighting using quadratic code length*, *dans* IEEE Computer Society International Conference on Computer Vision (ICCV)2013, IEEE, 2013, p. 1441–1448.

[43] N. Martin et S. Roy, *A comparison of coded light methods for precise and robust active reconstruction*, *dans* IEEE Transactions on Pattern Analysis and Machine Intelligence.2014, IEEE, 2014. Manuscript submitted for publication.

[44] M. Maruyama et S. Abe, *Range sensing by projecting multiple slits with random cuts*, IEEE Transactions on Pattern Analysis and Machine Intelligence., 15 (1993), p. 647–651.

[45] M. Mimou, T. Kanade et T. Sakai, *A method of time-coded parallel planes of light for depth measurement*, Transactions of the Institute of Electronics and Communication Engineers of Japan, E64 (1981), p. 521–528.

[46] T. Miyasaka, K. Kuroda, M. Hirose et K. Araki, *High speed 3-d measurement system using incoherent light source for human performance analysis*, *dans* International Society for Photogrammetry and Remote Sensing. ISPRS 2000., Jan 2000.

[47] R. Morano, C. Ozturk, R. Conn, S. Dubin, S. Zietz et J. Nissano, *Structured light using pseudorandom codes*, IEEE Transactions on Pattern Analysis and Machine Intelligence., 20 (1998), p. 322–327.

[48] D. Moreno et G. Taubin, *Simple, accurate, and robust projector-camera calibration*, *dans* 3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on, IEEE, 2012, p. 464–471.

[49] S. G. Narasimhan, S. J. Koppal et S. Yamazaki, *Temporal dithering of illumination for fast active vision*, *dans* European Conference on Computer Vision. ECCV 2008, Springer-Verlag, Jan 2008.

[50] S. K. Nayar, A. Krishnan, M. D. Grossberg et R. Raskar, *Fast separation of direct and global components of a scene using high frequency illumination*, ACM Transactions on Graphics, 25 (2006), p. 935–944.

[51] J. Pages, J. Salvi et J. Forest, *A new optimised de bruijn coding strategy for structured light patterns*, *dans* Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 4, aug. 2004, p. 284 – 287 Vol.4.

[52] P. Peers, D. K. Mahajan, B. Lamond, A. Ghosh, W. Matusik, R. Ramamoorthi et P. Debevec, *Compressive light transport sensing*, vol. 28, New York, NY, USA, February 2009, ACM, p. 3 :1–3 :18.

[53] J. Posdamer et M. Altschuler, *Surface measurement by space-encoded projected beam systems*, Computer Graphics and Image Processing, (1982).

[54] M. Proesmans, L. Van Gool et A. Oosterlinck, *One-shot active 3d shape acquisition*, International Conference on Pattern Recognition. ICPR 1996, 3 (1996), p. 336 – 340 vol.3.

[55] S. Roy, J. Meunier et I. J. Cox, *Cylindrical rectification to minimize epipolar distortion*, *dans* IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 1997, 1997, p. 393–399.

[56] J. Salvi, J. Batlle et E. Mouaddib, *A robust-coded pattern projection for dynamic 3d scene measurement*, Pattern Recognition Letters, 19 (1998), p. 1055 –1065.

[57] J. Salvi, S. Fernandez, T. Pribanic et X. Llado, *A state of the art in structured light patterns for surface profilometry*, Pattern Recognition., 43 (2010), p. 2666 – 2680.

[58] J. Salvi, J. Pages et J. Batlle, *Pattern codification strategies in structured light systems*, Pattern Recognition., (2004).

[59] T. Sato, *Multispectral pattern projection range finder*, *dans* Proceedings of SPIE, Jan 1999.

[60] D. Scharstein et R. Szeliski, *High-accuracy stereo depth maps using structured light*, *dans* IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2003, vol. 1, jun. 2003, p. 195–202.

[61] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein et R. Szeliski, *A comparison and evaluation of multi-view stereo reconstruction algorithms*, *dans* Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on, vol. 1, IEEE, 2006, p. 519–528.

[62] S. M. Seitz, Y. Matsushita et K. N. Kutulakos, *A theory of inverse light transport*, *dans* IEEE Computer Society International Conference on Computer Vision (ICCV)2005, vol. 2, IEEE, 2005, p. 1440–1447.

[63] P. Sen, B. Chen, G. Garg, S. R. Marschner, M. Horowitz, M. Levoy et H. Lensch, *Dual photography*, ACM Transactions on Graphics (TOG), 24 (2005), p. 745–755.

[64] V. Srinivasan, H. Liu et M. Halioua, *Automated phase-measuring profilometry of 3-D diffuse objects*, Applied Optics, (1984).

[65] R. Szeliski, *Computer Vision : Algorithms and Applications*, Springer-Verlag New York, Inc., Jan 2010.

[66] J. P. Tardif, S. Roy et M. Trudeau, *Multi-projectors for arbitrary surfaces without explicit calibration nor reconstruction*, International Conference on 3-D Digital Imaging and Modeling, (2003), p. 217–224.

[67] C. J. Taylor, *Implementing high resolution structured light by exploiting projector blur*, Applications of Computer Vision, IEEE Workshop on, 0 (2012), p. 9–16.

[68] M. Trobina, *Error model of a coded-light range sensor*, rap. tech., Marjan Communication Technology Laboratory Image Science, Jan 1995.

[69] P. Vuylsteke et A. Oosterlinck, *Range image acquisition with a single binary-encoded light pattern*, IEEE Transactions on Pattern Analysis and Machine Intelligence., 12 (1990), p. 148–164.

[70] G. Wetzstein et O. Bimber, *Radiometric compensation through inverse light transport*, *dans* Computer Graphics and Applications, 2007. PG '07. 15th Pacific Conference on, Oct 2007, p. 391–399.

[71] Y. Wexler, A. W. Fitzgibbon et A. Zisserman, *Learning epipolar geometry from image sequences*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2003, 2 (2003), p 209.

[72] Wikipédia, *Capteur photographique — wikipédia , l'encyclopédie libre.* `http://fr.wikipedia.org/w/index.php?title=Capteur_photographique&oldid=55488559`, 2010.

[73] Wikipédia, *Anticrénelage — wikipédia, l'encyclopédie libre.* `http://fr.wikipedia.org/w/index.php?title=Anticr%C3%A9nelage&oldid=92116829`, 2013.

[74] Wikipédia, *Correction gamma — wikipédia , l'encyclopédie libre.* `http://fr.wikipedia.org/w/index.php?title=Correction_gamma&oldid=90018536`, 2013.

[75] C. Wust et D. W. Capson, *Surface profile measurement using color fringe projection*, Machine Vision and Applications., 4 (1991), p. 193–203.

[76] Y. Xu et D. G. Aliaga, *An adaptive correspondence algorithm for modeling scenes with strong interreflections*, IEEE Transactions on Visualization and Computer Graphics, 15 (2009), p. 465–480.

[77] L. Zhang, B. Curless et S. Seitz, *Rapid shape acquisition using color structured light and multi-pass dynamic programming*, *dans* International Symposium on 3D Data Processing Visualization and Transmission. 3DPVT 2002, 2002, p. 24 – 36.

[78] S. Zhang et P. Huang, *Novel method for structured light system calibration*, Optical Engineering, (2006).

[79] S. Zhang et S. Yau, *High-speed three-dimensional shape measurement system using a modified two-plus-one phase-shifting algorithm*, Optical Engineering, (2007).

[80] S. Zhang et S.-T. Yau, *Generic nonsinusoidal phase error correction for three-dimensional shape measurement using a digital video projector*, Applied optics, 46 (2007), p. 36–43.

[81] H. Zhao, W. Chen et Y. Tan, *Phase-unwrapping algorithm for the measurement of three-dimensional object shapes*, Applied Optics, 33 (1994), p. 4497–4500.