Université de Montréal

Scanner 3D à lumière non structurée non synchronisé

par

Chaima El Asmi

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

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

août, 2018

RÉSUMÉ

Ce mémoire s'intéresse au domaine de la vision par ordinateur, et plus spécifiquement aux méthodes de reconstruction actives. Premièrement, il propose une nouvelle approche pour réaliser une correspondance par lumière structurée en résolvant le problème de la synchronisation caméra-projecteur. La carte de correspondances obtenue entre la caméra et le projecteur permet ensuite d'obtenir des modèles 3D denses et de grande qualité. Cette approche est idéale pour effectuer des scans de visages, qui sont difficiles à cause des mouvements involontaires des sujets. Deuxièmement, il propose une amélioration à la précision des correspondances en alignant, à une haute précision sous-pixel, le voisinage respectif de chaque pixel. Troisièmement, ce mémoire propose une application pratique de la méthode.

Jusqu'à maintenant, une grande attention était nécessaire pour s'assurer que chaque image de la caméra correspondait exactement au bon patron d'une séquence projetée. Ceci était très difficile à réaliser avec du matériel à faible coût ou dans des systèmes de capture à plus grande échelle. Dans notre méthode, un flux vidéo, constitué d'un nombre choisi de patrons de lumière non structurée, est projeté en boucle à une fréquence élevée (30 à 60 fps pour du matériel ordinaire) et ces patrons sont capturés par une caméra sans aucune forme de synchronisation. La seule contrainte à satisfaire est que la fréquence de capture de la caméra et la fréquence de projection du projecteur soient connues. Le processus de correspondance permet, non seulement, de retrouver la bonne séquence de patrons, mais est aussi robuste aux expositions partielles de deux patrons consécutifs en plus de l'effet d'obturateur déroulant (*rolling shutter*). Par ces étapes, le problème important de la synchronisation caméra-projecteur est pris en charge.

L'amélioration de la précision des correspondances est obtenue par le recours à l'interpo-

lation sous-pixel des intensités des images suivant la mise en correspondance. On obtient ainsi une méthode comparable aux meilleures approches de mise en correspondance sous-pixel, mais avec une plus grande robustesse. Les résultats obtenus illustrent des modèles 3D reconstruits avec une très haute précision dans des conditions difficiles telles que l'illumination indirecte, les scènes discontinues ou l'utilisation d'un matériel non professionnel.

La dernière partie de ce mémoire se consacre à la description de la conception et la fabrication d'un casque de réalité virtuelle personnalisé épousant les contours du visage de l'utilisateur. Avec la méthode dense et précise préalablement proposée, la méthode à lumière non structurée non synchronisée sous-pixel, il est possible d'obtenir à très bas coût un casque sur-mesure et confortable.

Mots clés: Vision par ordinateur, Reconstruction active, Lumière non structurée, Système caméra-projecteur non synchronisé, Précision sous-pixel, Scanner 3D.

ABSTRACT

This thesis is in the field of computer vision and specifically focuses on active reconstruction methods. Firstly, it proposes a new approach to structured light correspondence which alleviates the camera-projector synchronization problem. This allows the creation of a dense correspondence map between the camera and projector which can further be used to obtain 3D models. This high-speed approach is ideal for scanning human faces, which are notoriously difficult because of involuntary motions of the subjects. Secondly, it proposes an improvement to the matching accuracy by aligning, to high subpixel precision, the respective neighborhood patches of each pixel. Thirdly, an example of an application of the method is proposed.

Until now, great care was required to make sure that each camera image was captured exactly when the correct pattern was projected. This was difficult to achieve with low-cost hardware or large size installations. In our method, the projector sends a video loop of a selected number of unstructured light patterns at a high frame rate (30 to 60 fps for common hardware), which are captured by a camera without any form of synchronization. The only constraint to satisfy is that the camera and projector frame rates are known. The matching process not only recovers the correct pattern sequence, but is robust to partial exposures of consecutive patterns as well as rolling shutter effects. Through these steps, the crucial camera-projector synchronization problem is entirely taken care of.

Subpixel matching accuracy is obtained by relying on image interpolation to refine the original correspondences. These results are highly accurate and comparable to other state-of-the-art subpixel methods but with better robustness. The obtained results illustrate 3D models reconstructed with very high precision in difficult conditions such as indirect illumination, scene discontinuities, and use of low quality hardware.

The last part of this thesis is devoted to the description of the design and manufacturing of a customized virtual reality headset which snuggly matches a user's face, thanks to 3d models constructed from the matching method proposed in the thesis, where high quality is obtained on low-cost hardware.

Keywords : Computer vision, Active reconstruction, Unstructured light, Unsynchronized camera-projector systems, Subpixel accuracy, 3D scanning.

TABLE DES MATIÈRES

LISTE DES FIGURES

REMERCIEMENTS

Je tiens tout d'abord à remercier mon directeur de recherche, Sébastien Roy, pour m'avoir donné l'opportunité de réaliser ma maîtrise. Je le remercie pour tout ce qu'il m'a apporté durant ces dernières années à travers toutes les discussions enrichissantes, pour son enthousiasme et son énergie.

Je tiens à remercier tous les membres du laboratoire que j'ai eu le plaisir de côtoyer pendant ces années, plus particulièrement Pierre-André qui m'a apporté un énorme soutien et une aide considérable et précieuse.

Je remercie finalement ma famille pour m'avoir donné la force de continuer, plus particulièrement mes parents pour leur soutien et leur encouragement malgré la distance, et ma sœur pour sa présence et ses conseils.

INTRODUCTION

Ce mémoire s'intéresse à deux problèmes majeurs dans le domaine des reconstructions actives. Les travaux présentés ont pour but d'améliorer la méthode à lumière non structurée afin de pouvoir scanner des visages rapidement avec du matériel ordinaire.

En premier lieu, nous nous sommes attaqués au problème de synchronisation entre le projecteur et la caméra. En utilisant du matériel à faible coût et en projetant à une fréquence d'images élevée (30 fps et 60 fps), il est possible de réaliser des scans 3D d'objets en moins de deux secondes sans synchroniser le système projecteur-caméra. À l'aide de cette méthode, les résultats obtenus sont démontrés très proches de ceux obtenus avec la capture synchronisée. Cette méthode permet de faciliter et d'accélérer le scan des visages ainsi que les scans dans des systèmes de caméras et de projecteurs très éloignés où il serait habituellement impossible de les synchroniser. De plus, cette méthode se veut accessible aux utilisateurs qui n'ont pas accès à du matériel industriel ou de fine pointe. Elle leur permet d'obtenir facilement des modèles 3D denses de qualité.

En deuxième lieu, nous avons visé à améliorer la qualité des correspondances en ajoutant du sous-pixel à l'algorithme de correspondance. Le sous-pixel est un facteur important dans la reconstruction 3D, car il augmente la précision et le niveau de détails. De plus, le sous-pixel permet de corriger les erreurs de correspondances afin d'obtenir une qualité de correspondance supérieure. Dans la méthode proposée, l'estimation du sous-pixel est facile et rapide. Nous avons obtenu des modèles 3D à haute précision dans des conditions difficiles tels que l'illumination indirecte, les scènes discontinues ou l'utilisation d'un matériel non professionnel. Il a été possible de comparer les modèles réalisés à d'autres modèles réalisés par des méthodes classiques. Aussi, une évaluation est effectuée sur différents paramètres qui affectent le sous-pixel tels que le ratio de pixels et la fréquence spatiale des

patrons.

En troisième lieu, la méthode proposée offre une robustesse et une précision équivalentes sinon meilleures que les méthodes classiques, mais avec une plus grande rapidité et robustesse. Les scanners 3D sont omniprésents dans divers domaines depuis plusieurs années. Ainsi, nous pensons que ce scanner 3D se compare avantageusement avec ceux disponibles sur le marché, et ce à cause de leur prix onéreux ou de leur longue durée de scan. Il existe diverses applications aux scanners 3D, l'application à l'étude dans ce mémoire est le casque personnalisé de réalité virtuelle, qui épouse parfaitement les contours du visage grâce au scan 3D rapide du visage. Avec le modèle précis du visage qui est obtenu, il est d'ailleurs possible de fabriquer et imprimer un casque VR sur mesure qui procure un grand confort.

Plusieurs perspectives de recherches futures restent à explorer dans le domaine des scanners 3D. L'une de ces perspectives touchant notre approche à lumière non structurée non synchronisée sous-pixel est le scanner à main libre. Notre matériel actuel est attaché sur un support amovible et doit rester immobile pendant au moins deux secondes afin de réussir à scanner des scènes statiques. Dans le futur, il est envisageable de mettre au point un scanner 3D qui peut scanner des scènes dynamiques ou scanner "main libre".

Chapitre 1

MÉTHODES DE RECONSTRUCTION ACTIVES

La reconstruction consiste à produire un modèle 3D à partir de correspondances denses souvent sous la forme de nuages de points. Ces correspondances sont obtenues à partir de deux images prises de deux points de vue différents d'un même objet où chaque pixel de l'objet est ensuite déprojeté à une profondeur. Les pixels de l'objet dans la première image sont mis en correspondance avec les pixels du même objet dans la seconde. Cette méthode n'interagit pas avec la scène dans la mesure autre que la capture de photon, c'est une méthode de reconstruction dite passive. Toutefois, cette méthode est plus faible dans la reconstruction de régions uniformes ou similaires. Elle génère une ambiguïté de mise en correspondance [30, 8] (tels que les intensités similaires ou les motifs répétitifs). Une reconstruction obtenue à l'aide de cette méthode n'est alors pas qualifiée de dense. De plus, cette méthode nécessite que les caméras soient calibrées pour obtenir une carte de profondeur [43].

1.1 Méthodes à lumière structurée

Les méthodes de reconstruction actives viennent apporter une solution au problème de surfaces sans textures. En ajoutant de l'information supplémentaire sur la scène à reconstruire, cela augmente les correspondances entre les pixels. Afin de convertir un système passif vers un système actif, il suffit de remplacer l'une des caméras en projecteur puis projeter de la lumière codée sur un objet qui peut maintenant être uniforme [44] (par exemple un mur blanc). Dans ce cas, le projecteur représente le dispositif actif. Il projette les patrons à lumière codée (*Coded patterns*) et la caméra capture les transformations obtenues. Les patrons à lumière codée représentent des images qui encodent des motifs pour faciliter la

correspondance. Pour chaque pixel dans l'image capturée par la caméra, le code est défini comme étant la position du pixel correspondant dans l'image du projecteur. Autrement dit, une longueur de code correspond à l'ensemble de la séquence. La méthode naïve pour identifier chaque pixel dans le projecteur consiste donc à les allumer, un à la fois, l'un après l'autre. De cette manière, il est possible d'associer directement chaque pixel de la caméra à son correspondant dans le projecteur. Cependant, cette méthode requiert énormément de temps, car elle encode des milliers de pixels, un à la fois. Afin de pallier à cette faiblesse, d'autres méthodes se sont inspirées de cette technique en améliorant les patrons. Les méthodes à lumière structurée encodent directement la position du pixel dans l'image capturée à l'aide de la projection de patrons. La classification de ces méthodes se base sur la manière de générer les patrons.

1.1.1 Méthodes non temporelles

Ces méthodes sont parfois aussi nommées méthodes spatiales, car elles se basent sur le voisinage. Le code d'un pixel est retrouvé à partir de son voisinage. Ce type de méthodes ne requiert pas beaucoup de patrons. En effet, il est possible d'obtenir des correspondances à l'aide d'un seul patron projeté. Les méthodes non temporelles sont utilisées généralement pour les scènes dynamiques puisqu'elles fonctionnent en temps réel. L'une des méthodes de ce type utilise les *séquences de De Bruijn*. Les patrons sont générés en utilisant des séquences semi-aléatoires. Ils sont composés de plusieurs bandes de couleurs pour pouvoir retrouver le voisinage [17, 27], comme le montre la Fig. 1.1. De plus, deux bandes consécutives ne peuvent pas avoir les mêmes couleurs. Cette méthode est très utilisée pour scanner des scènes dynamiques [48]. La faiblesse principale de ces méthodes est la présence de discontinuités dans la scène. La caméra n'est pas toujours capable d'observer toutes les bandes [48]. Ce type de méthodes assume une continuité spatiale de la scène, ce qui n'est pas toujours le cas. Cela rend l'étape de décodage très difficile et cela induit beaucoup d'erreurs. Une seconde faiblesse est que les patrons sont unidimensionnels; ils

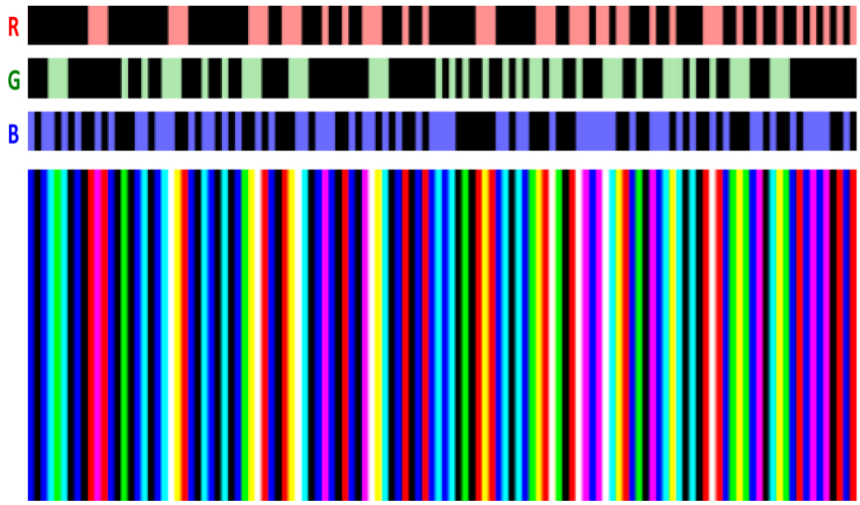


Figure 1.1: Les patrons de De Bruijn [3] constitués de séquences semi-aléatoires. Ils sont composés de plusieurs bandes de couleurs {R,G,B} afin de pouvoir retrouver le voisinage d'un pixel donné.

encodent un seul axe à la fois. Finalement, le système caméra-projecteur doit absolument être calibré pour permettre la reconstruction d'un objet en 3D.

1.1.2 Méthodes temporelles

Contrairement aux méthodes spatiales, les méthodes temporelles requièrent beaucoup de patrons. Elles se basent sur la projection d'un ensemble de patrons successifs dans le temps. C'est pourquoi la scène doit être statique durant l'entièreté du scan. Le code est formé par la séquence des intensités de tous les patrons projetés. Il est important que la caméra voie chaque patron indépendamment une fois.

Dans [39], les auteurs utilisent une séquence de patrons binaires pour générer un code binaire. Il y a uniquement deux intensités dans les patrons; blanche ou noire qui correspondent à 1 et 0 respectivement. Ils proposent de projeter une séquence de n patrons pour encoder 2^n bandes. La séquence de code est formée de 0 et de 1. La propriété de cette méthode est que le nombre de bandes noires et blanches augmente d'un facteur 2 entre deux

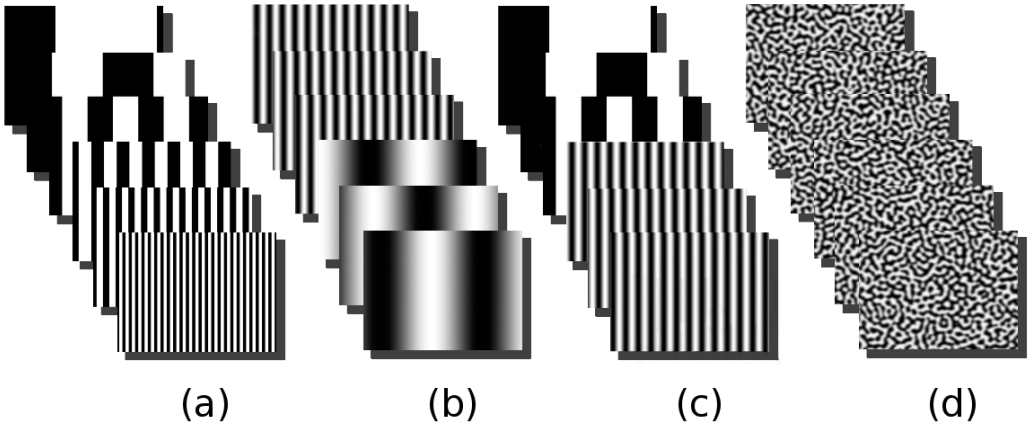


Figure 1.2: De droite à gauche, des exemples de patrons à lumière codée qui représentent (a) la méthode de *Code de Gray*, (b) la méthode de *Phase Shift*, (c) la méthode hybride: *Code de Gray* et *Phase Shift* combinés et enfin (d) la méthode à lumière non structurée.

patrons consécutifs. Avec cette méthode, les patrons sont unidimensionnels. Il faut donc encoder chaque axe à la fois pour obtenir une position en x et en y. Il est parfois difficile, lors de l'étape de décodage, de déterminer si l'intensité du pixel est blanche ou noire. La solution est donc de projeter un patron et son inverse. Ce qui double le nombre de patrons à projeter. De plus, au moment de la transition entre deux codes voisins, il est possible que tous les bits soient différents. Ce changement peut induire beaucoup d'erreurs et engendrer une correspondance différente.

Dans [26], les auteurs ont développé une variation du code binaire afin de minimiser l'effet des erreurs de bits. En effet, deux codes consécutifs ont une *distance de Hamming* égale à 1. En d'autres termes, deux codes consécutifs ont seulement un seul bit de différence. La Fig. 1.2 (a) représente les patrons de *Code de Gray*. C'est une méthode très robuste au bruit. Néanmoins, elle est faible à l'illumination indirecte à cause de la fréquence de ces patrons (les bits de poids forts). De plus, les bits de poids faibles peuvent être perçus comme du gris par la caméra. Les méthodes présentées ci-haut sont dites méthodes discrètes et ainsi elles ont une résolution assez limitée.

Dans [53], une nouvelle approche a été utilisée en projetant le même patron projeté plusieurs fois, mais décalé dans une certaine direction. La Fig. 1.2 (b) illustre les patrons de cette méthode. Ils sont constitués d'un déphasage de motifs sinusoïdaux. Le simple fait de projeter une intensité périodique plutôt que binaire permet de hausser la qualité des correspondances et d'améliorer la résolution. Chaque intensité est reliée à la phase du pixel du projecteur. À partir des phases récupérées, il suffit de retrouver la période pour avoir la correspondance entre les pixels. Or, la nature périodique des patrons introduit une ambiguïté au moment de déterminer les périodes du signal dans la séquence capturée. Il existe une solution basée sur l'algorithme de désambiguïsation de la phase (*phase unwrapping*) présentée dans les articles [29, 59]. L'idée est de projeter plusieurs sinus de différentes fréquences, chacun des sinus est déphasé plusieurs fois. Ainsi, il y a moyen de retrouver la période de chaque phase.

D'autres méthodes ont intégré le *Code de Gray* avec le *Phase Shift* [9] afin d'enlever l'ambiguïté des périodes. En effet, la position du pixel est résolue par le *Code de Gray*. Cela permet de retrouver la période correspondante. Cette méthode hybride, illustrée dans la Fig. 1.2 (c), a l'avantage d'avoir un code robuste et un scan à haute résolution. Malgré la robustesse et la haute résolution, ces méthodes souffrent de l'illumination indirecte ainsi que la discontinuité de la scène.

D'autre part, il est possible qu'un seul pixel de la caméra capture une combinaison linéaire de deux pixels ou plus adjacents du projecteur à cause de la résolution ou la géométrie relative. C'est ce que nous appelons le ratio de pixels caméra-projecteur. Les méthodes à lumière structurée se dégradent plus avec le ratio de pixels. Nous pouvons citer comme exemple les bits de poids faible de la méthode *Code de Gray* qui deviennent flous pour être détectés par la caméra. Ces trois facteurs engendrent énormément d'erreurs au niveau des correspondances.

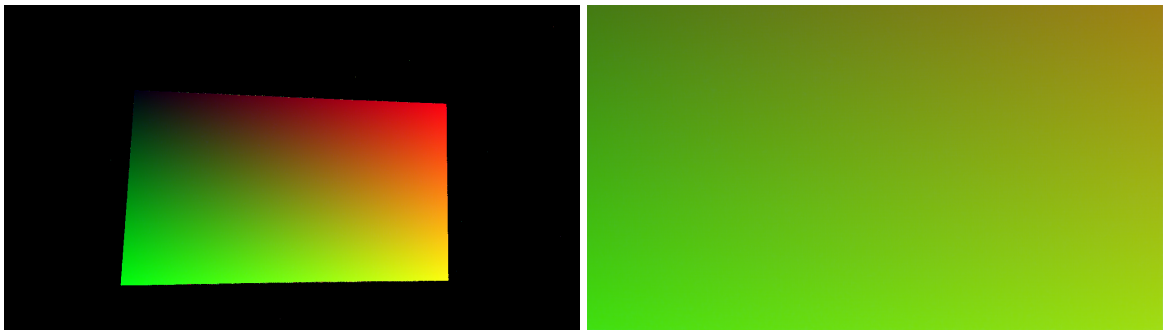


Figure 1.3: La correspondance à lumière non structurée est bidirectionnelle. La figure (à droite) représente la correspondance de caméra vers projecteur (*LUT* du point de vue de la caméra) et la figure (à gauche) représente la correspondance du projecteur vers caméra (*LUT* du point du projecteur). Les couleurs rouge et verte représentent les positions x,y, respectivement.

1.2 Méthodes à lumière non structurée

Les méthodes à lumière non structurée, contrairement à celles par lumière structurée, n'encodent pas directement la position du pixel [12, 32, 52, 14]. Ainsi, les patrons ne dépendent pas de la position du pixel du projecteur. D'ailleurs, ces méthodes ne requièrent aucune forme de calibrage. Les auteurs de [32] sont les premiers qui ont obtenu des correspondances en projetant du bruit aléatoire. Le code binaire est obtenu à partir de la séquence temporelle projetée qui est par la suite mise en correspondance avec la séquence de référence. La position de chaque pixel et son correspondant sont encodés dans une *LookUp-Table* (LUT). La Fig. 1.3 représente des correspondances encodées dans des LUTs du point de vue de la caméra et du projecteur.

Prenons la vue de la caméra, les pixels de la LUT représentent la position de chaque point de la caméra dans le monde. Ensuite, les couleurs représentent la position de chaque point correspondant dans le projecteur (la couleur rouge pour l'axe x, la couleur verte pour l'axe y et la couleur bleue pour le taux d'erreur).

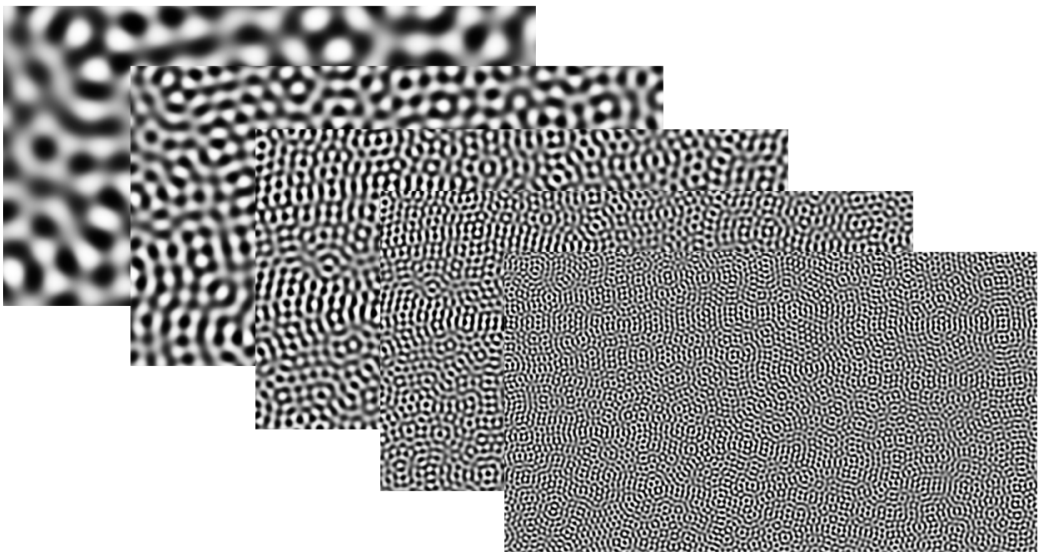


Figure 1.4: Patrons à lumière non structurée générés avec différentes fréquences ({15, 30, 50, 70, 100} de gauche vers droite respectivement).

Dans [12], ils améliorent les patrons à projeter tel qu'illustré dans la Fig. 1.2 (d). Ces nouveaux patrons sont conçus pour minimiser l'effet de l'illumination indirecte. Les patrons sont générés dans le domaine fréquentiel en utilisant une *Transformée de Fourier*. Ensuite, ils appliquent un flou gaussien sur les patrons binarisés afin d'équilibrer la quantité de blanc et de noir dans l'image. De cette manière, les grandes zones blanches ou noires sont grandement réduites dans les patrons. Cette méthode est donc robuste à l'éclairage indirect et aux différents ratios de pixels. Avec un nombre de patrons raisonnables, les auteurs obtiennent des correspondances denses dans des conditions difficiles de capture ainsi que dans des scènes discontinues. L'autre avantage de la projection à lumière non structurée est la bidirectionnalité des correspondances. En effet, il est possible d'encoder les pixels de la caméra vers le projecteur ou en sens inverse (du projecteur vers la caméra).

Dans cette méthode, un autre facteur à considérer est la fréquence qui joue un rôle important. La qualité des correspondances dépend de la fréquence spatiale ainsi que le nombre de bits de code (le nombre de patrons). Lorsque la fréquence des patrons est trop

basse, les codes voisins d'un pixel deviennent trop similaires et cela engendre des erreurs dans la mise en correspondance. Il est possible de remédier à ce problème en augmentant soit le nombre de patrons soit la fréquence. Inversement, lorsque la fréquence est très haute, il apparaît le risque que la caméra ne soit plus capable de distinguer les zones blanches et noires. Elle ne voit alors qu'un mélange de gris.

Au lieu d'augmenter le nombre de patrons, les auteurs, dans [34], ont augmenté la longueur de code en regardant le signe de la différence entre les intensités et l'image moyenne. Ainsi, un code linéaire est généré en comparant chaque patron avec l'image moyenne. Pour n patrons, ils obtiennent une longueur de code égale à n pour n bits d'information. Ils ont proposé une nouvelle méthode pour générer le code afin d'augmenter la quantité d'informations. Cette méthode considère toutes les paires possibles des patrons et de cette façon un code quadratique génère une longueur de code égale à $\frac{n^2-n}{2}$ pour $n \log n$ bits d'information.

La propriété aléatoire des patrons va nous aider à résoudre le problème de synchronisation, expliqué dans le chapitre suivant. Dans ce travail, la méthode de génération des patrons diffère, mais certaines de leurs propriétés sont conservées. Nos patrons ne présentent pas de larges régions blanches ni noires. Ils sont chacun une somme de sinus avec des orientations aléatoires, mais une même fréquence de sorte que chaque patron à lumière non structurée est différent. La Fig. 1.4 représente une variété de fréquences de nos patrons.

Chapitre 2

SYNCHRONISATION

La synchronisation est un des systèmes de base d'un scanner 3D. Un scanner à lumière structurée ou non structurée est composé d'un projecteur et d'une caméra. Il faut projeter les patrons et les capturer pour finalement obtenir des correspondances denses ce qui est difficile sans synchroniser le système caméra-projecteur. En effet, le projecteur et la caméra doivent voir la même image projetée au même moment. Chaque patron projeté ne doit pas changer durant l'exposition de l'image ce qui peut ralentir l'acquisition des images.

Généralement, la projection synchronisée est supposée connue par les méthodes et n'est par conséquent pas détaillée. Les approches sont classées en deux grandes catégories. La première est la synchronisation matérielle. Elle consiste à utiliser un dispositif qui synchronise le déclenchement de l'acquisition caméra et de la projection des patrons. Dans [50, 21, 41], ils utilisent une vidéo *genlock* pour synchroniser la caméra et pouvoir capturer chaque patron indépendamment. D'autres méthodes utilisent une caméra avec une entrée de déclenchement externe. Un signal électrique est envoyé par le projecteur ou par un *triggering circuit* afin de déclencher la caméra [40, 27, 49, 33, 51]. Ce type de méthodes peuvent atteindre des performances d'acquisitions synchronisées allant jusqu'à 3000 fps. Toutefois, ces méthodes requièrent du matériel spécialisé tel que des projecteurs qui ont des entrées ou des sorties de connexion pour pouvoir synchroniser le système caméra-projecteur. Ce matériel est difficile à obtenir par des particuliers puisque souvent il est développé en laboratoire pour faire l'objet de publication ou en industrie et est sujet à une distribution plus limitée.

La deuxième approche consiste à synchroniser le système caméra-projecteur à l'aide d'un logiciel, ce qui requiert que le projecteur et la caméra soient tous deux connectés à un ordinateur. Dans [28], chaque patron doit être mis en pause pendant quelques secon-

des afin que la caméra puisse exposer entièrement. Bien que les caméras puissent capturer à de hautes vitesses d'images par seconde, il est difficile d'évaluer de façon fiable le temps d'attente nécessaire entre le déclenchement et la capture d'un patron. Ceci mène donc à surévaluer les temps d'attentes pour assurer une bonne capture de chaque patron. Dans [24], ils utilisent un système qui attend pendant un certain intervalle de temps entre la projection et l'acquisition des images. Toutefois, cette approche augmente le temps d'acquisition. De plus, le scan est ralentit encore plus avec les caméras *rolling shutter*. Ces dernières exposent une seule ligne à la fois. Ces méthodes fonctionnent donc seulement à des fréquences d'images très basses.

Que ce soit synchroniser le système caméra-projecteur avec un logiciel ou un matériel, ces deux approches présentent énormément d'inconvénients. D'un côté, le matériel coûte très cher et il n'est pas très accessible au public. Généralement, c'est un équipement expérimental industriel. De l'autre, il est possible d'utiliser du matériel commun, mais le scan requiert un temps d'acquisition trop long. L'absence de synchronisation apparaît donc comme une solution pour certains. Il existe des méthodes telles que la *Microsoft Kinect* [58], *Google Project Tango* [2] et *Sensor Structure* [4] qui sont considérés comme étant des scanners non synchronisés. C'est des patrons statiques qui ne nécessitent pas d'être synchronisés. Récemment, une méthode de scan à lumière structurée non synchronisé à vue le jour [35]. Cette méthode se veut très proche de la solution que nous proposons. Les auteurs ont résolu un modèle de formation de l'image afin de retrouver les paramètres de synchronisation. À l'aide de ce modèle de formation de l'image, ils déterminent le temps requis par la caméra pour capturer chaque ligne projetée. Ensuite, ils calculent le temps d'exposition d'une image entière d'une caméra *rolling shutter*. De cette manière, ils sont capables de résoudre le problème d'exposition multiple causée par la non-synchronisation. Afin de retrouver la première image de la séquence capturée, ils projettent une séquence d'images blanches et noires au début de la séquence. Au moment de décodage, ils peuvent facilement déterminer le début de la séquence. Il est à noter qu'ils n'utilisent pas de l'équipement couramment utilisé. Ils utilisent un projecteur spécial tel qu'un *DLP*

LightCrafter. Les faiblesses de cette méthode sont qu'elle demande un calcul complexe à cause du système d'équations à résoudre ainsi que beaucoup de temps de calcul.

Dans le prochain chapitre, nous présentons une nouvelle approche de non-synchronisation. Aussi, nous redéfinissons le terme de synchronisation. Nous allons expliquer trois aspects importants, selon nous, de la non-synchronisation (Fig. 3.1). Le premier aspect est comment retrouver la première image dans la séquence capturée. L'objectif est de projeter une vidéo constituée de patrons à lumière non structurée en boucle et capturer à n'importe quel moment. Ainsi, la première image est inconnue dans la séquence capturée et doit être trouvée. Si la séquence capturée n'a pas le même ordre que la séquence de référence, nous ne pourrons pas récupérer les correspondances entre la caméra et le projecteur. Pour simplifier ce problème, dans [28], ils répètent la projection du premier patron plusieurs fois ainsi à l'étape de décodage, il est très facile de la retrouver. Dans notre cas, cela n'est pas nécessaire. Le deuxième aspect concerne le décalage temporel. Le processus de projeter et de capturer non synchronisé à une haute fréquence d'images, engendre un décalage temporel ce qui entraîne un mélange dans l'image capturée entre deux patrons consécutifs. De plus, l'utilisation de caméra *rolling shutter* (exposition multiple) rend l'étape de décodage encore plus difficile. Le dernier aspect est la fréquence d'images de la caméra et du projecteur. S'ils ne sont pas pareils ou ils ont varié au cours du temps, cela modifie la séquence capturée. Il peut y avoir plusieurs images dupliquées d'un seul patron ou alors des patrons manquants. Dans notre cas, nous présumons que les fréquences d'images de la caméra et du projecteur sont connues à l'avance et ne changent pas au cours du temps. En pratique, il est très facile de les fixer à l'avance.

Dans le chapitre suivant, nous expliquons plus en détail le fonctionnement de notre méthode sous forme d'un article. C'est une méthode de scan à lumière non structurée non synchronisé simple et rapide. Elle est adaptée pour les scènes telles que des visages qui ne peuvent pas rester stables plus que deux secondes.

Chapitre 3

(ARTICLE) FAST UNSYNCHRONIZED UNSTRUCTURED LIGHT

Cet article [16] a été publié tel qu'indiqué dans la bibliographie :

> C. El Asmi et S. Roy. Fast unsynchronized unstructured light. Computer and Robot Vision (CRV), 2018 15th Conference on, (2018).

Dans cet article, nous présentons une nouvelle approche au problème de synchronisation entre une caméra et un projecteur, en utilisant un matériel pas cher. Chaque image projetée doit être vue une seule fois par la caméra. Le projecteur projette une vidéo en boucle qui contient un certain nombre de patrons de lumière non structurée (30 ou 60 images par seconde). La caméra commence à capturer à tout moment. La projection à lumière non structurée non-synchronisée engendre des problèmes dans la séquence capturée. La première image dans la séquence capturée ne correspond pas nécessairement au premier patron projeté. Ainsi, il faut réordonner la séquence capturée. L'utilisation d'un matériel pas cher (tel que des caméras *rolling shutter*) et la capture à 30 images par seconde engendrent une exposition multiple entre deux patrons de lumière non structurée consécutives. Notre méthode consiste à retrouver le mélange entre ces deux patrons. Toutes les méthodes synchronisées utilisent de la lumière structurée. Afin de comparer la précision et la qualité du non-synchronisé objectivement, nous avons projeté de la lumière non structurée synchronisée et non-synchronisée.

L'article est présenté dans sa version originale avec des modifications mineures.

Abstract

This paper proposes a new approach in structured light correspondence to alleviate the camera-projector synchronization problem. Until now, great care was required to make sure that each camera image was corresponding exactly to the correct pattern in the sequence. This was difficult to achieve with low-cost hardware or large size installations. In our method, the projector sends a constant video loop of a selected number of unstructured light patterns at a high frame rate (30 to 60 fps for common hardware), which are captured by a camera without any form of synchronization. The only constraint to satisfy is that the camera and projector frame rates are known. The matching process not only recovers the correct pattern sequence, but is impervious to partial exposures of consecutive patterns as well as rolling shutter effects.

3.1 Introduction

The first appearance of 3D scanners goes back several decades. For some time now, researchers have been looking for a way to scan objects and get realistic 3D models. Nowadays, there are several 3D scanners based on different approaches to achieve 3D reconstruction. One is based on the projection of a laser beam and analyzes its trajectory as well as its deviation. Another one calculates how long a laser takes to get to the surface of the object and back to the projector, so-called *time-of-flight* [10]. Finally, some 3D scanner use structured light, which consists of projecting a known pattern encoding matching information onto an object and capturing so its geometry can be derived [46].

The most common method is to use a DLP projector and rely on Gray code patterns. They provide a temporal binary encoding of the position of each projector pixel. In this way, it is possible to establish the correspondence directly between the camera and the projector. However, a single error in a pattern will result in a wrong decoded position. A Gray code feature is that neighboring pixel codes differ only by a single bit. This spreads evenly the number of error bits accross all possible codes [26]. However, the presence

of low frequency spatial patterns in Gray code images increase the sensitivity to inter-reflections and indirect illumination [36, 19].

3.1.1 Unstructured Light

Unstructured light is also a projected encoding, but it does not directly encode the pixel positions [12, 32, 52, 14]. The temporal sequence provides a binary code which is then matched to a reference sequence. The decoupling of encoding and patterns makes it possible to select the spatial frequency and achieve great robustness to indirect lighting as well as difficult capture conditions. In this paper, the complete flexibility in the selection of unstructured patterns will prove essential to solve the synchronization problem. An important advantage is that it is possible to establish an unstructured light correspondence from the projector to the camera as well as the usual camera to the projector matching. This alleviates the need to compute an inverse mapping, which is prone to errors.

3.1.2 Synchronization

The synchronization of the camera-projector system can take multiple forms. In this section, we will describe the three most important aspects for our method: first image, temporal offset, and frame rate, as illustrated in Fig. 3.1. First, the projector sends a video loop repeating n patterns and the camera starts capturing at an arbitrary time. The first image in the sequence is therefore an unknown pattern and must be estimated (Fig. 3.1, first). To make this problem simple, some methods project special patterns, or repeat the first pattern, in order to allow easy identification of the start of the sequence [28]. This will not be necessary in our case. Secondly, there is a temporal offset between the camera and the projector image sequences (see Fig. 3.1, mix). This offset generally results in a mixture of two consecutive patterns in a single captured image. The multiple exposure problem makes the decoding or establishing correspondence between camera and projector images impossible. Aside from using synchronization, the multiple exposure problem can be solved with an

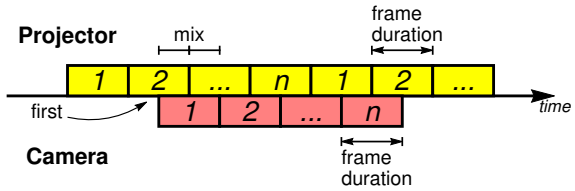


Figure 3.1: Synchronization model. The projector and camera share a common frame rate (fps). The first projector pattern seen by the camera can be any pattern. Exposure time and temporal overlap result in a mixture of two consecutive patterns in a single camera image.

accurately calibrated image formation model [35] or by designing the method to be robust to this effect, as we do in this paper. Finally, the last aspect is the frame rate of the camera and the projector (see Fig. 3.1, frame duration). Both frame rates are assumed to be known in advance, and to never change in time. It is possible to account for differences in frame rates, but the simplest and most practical approach with common hardware is to enforce the same frame rate for both the camera and projector, thereby ensuring stable multiple exposure of consecutive frames. When frame rates are not matched, we can end up with many duplicate images of a single pattern, or missing patterns. These two cases, even if they can be alleviated with some effort, are easy to avoid in practice, since frame rates tend to be standardized and stable.

In this article, we will explain our method, proceed to identify the first pattern, and then establish correspondence with the reference patterns while discovering the mix of multiple exposures.

3.2 Previous work

There has not been much work on the specific problem of unsynchronized structured light. Most methods assume perfect synchronization of the camera-projector system or try to improve it. The article [38] summarizes well the three main approaches of this problem: hardware synchronization, software synchronization, and no synchronization. The first one

is hardware synchronization. It is a triggering circuit that synchronizes the projector and the camera [27, 33, 51]. This approach is not accessible to everyone because it requires hardware, namely a projector and a camera that each supports external synchronization. This is rare for inexpensive hardware. Its advantage is that the system can scan at a very high frame rate (sometimes up to 3000 fps) [49, 55]. The second approach is to synchronize the camera-projector system by software [28, 37, 31]. Unlike the previous one, it does not cost as much and allows for using common off-the-shelf hardware, but it requires a longer acquisition time [24]. It must ensure that each projected pattern is captured by the camera without a risk of multiple exposure of consecutive patterns. The resulting scan time is not acceptable in many cases, such as scanning faces. In such situations, the acquisition time has to be reduced because a person cannot stay still more than a few seconds. Also, since we intend to use low-cost hardware, no synchronized solution is adequate. For these reasons, we propose the unsynchronized approach.

There are some methods that are based on the unsynchronized approach [35, 42, 58]. One successful method that is close to ours is [35], where a detailed image formation model is devised and solved to recover synchronization parameters and subsequently Gray code matching patterns. Their algorithm relies on exact knowledge of the image formation process to be able to determine the time required by the camera to capture each single row of the projected image (they assume a rolling shutter camera), as well as the time required to capture the entire image. In this way, they can resolve the multiple exposure problem present in the unsynchronized captured images. To detect the first projected pattern, extra frames are added at the beginning of the sequence. They are either full black (B) or full white (W) frames, set up as the sequence {B, B, W, W, B} so it becomes easy to detect it in the camera image sequence.

Overall, their method is quite complex computationally, especially given the equation systems that have to be resolved, and its reliance on Gray code patterns makes it more sensitive to inter-reflections. However, their results indeed prove that unsynchronized capture can be done, even if the hardware requirement is not completely general and require a

special fast projector.

In this paper, we present a very simple and fast method. Described in Sec. 3.3, it scans without synchronization at 30 to 60 fps in less than two seconds. Moreover, it does not require any special hardware, as the experiments presented in Sec. 3.4 will illustrate.

3.2.1 Generating unstructured light patterns

In this paper, we rely on the concept of unstructured light patterns presented by Couture *et al* [12]. However, the patterns are generated in a different way, such that they do not present large white or black regions. They are built simply as a sum of sines with random orientations but similar frequencies, so each unstructured light pattern is different. Fig. 3.2 shows two such patterns at different frequencies. The quality of the unstructured match depends on the spatial frequency and the number of bits of code, which itself depends on the number of patterns. If the frequency is too low, then the codes of neighbouring pixels become too similar which causes bad matches. Increasing the number of patterns can help remove this ambiguity. Using higher frequency patterns can also help (see Fig. 3.2, right), but if the frequency becomes too high then the camera might not be able to distinguish light and dark bands, resulting in matching errors.

3.2.2 Quadratic code

Our method will use linear code to find the first image of the sequence, and then quadratic code to estimate the mixture of the captured images and to compute the final matching.

As demonstrated by Couture *et al* [34], for n patterns, the linear code-length is n bits long and provides n bits of information. On the other hand, the quadratic code is derived from the same n patterns, has a length of $\frac{n^2-n}{2}$ bits, which provide $n \log n$ bits of information, the maximum possible out of n patterns. As an example, 30 patterns can provide a linear code of length 30 bits for 30 bits of information, or a quadratic code of length 435 bits for 147 bits of information.

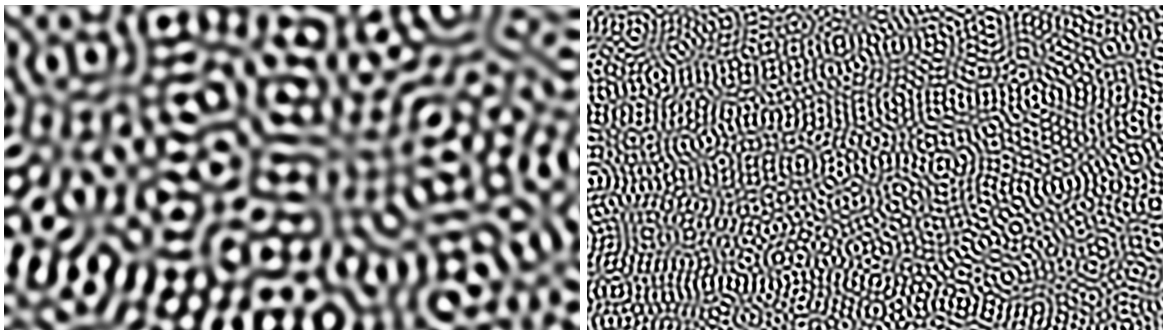


Figure 3.2: Unstructured light patterns can be chosen arbitrarily. Above, two such patterns at different spatial frequencies.

Establishing pixel correspondence between camera and projector is accomplished by the LSH (Locality Sensitive Hashing) algorithm [7]. This probabilistic algorithm is specialized in searching for nearest neighbors in very high-dimensional spaces. At each iteration, it generates different match proposals from which we keep only the best one at the end. Because of its probabilistic nature and efficiency in high dimensions, LSH is better suited to match long codes. For these reasons, in order to increase the amount of information and decrease the matching error, we prefer using quadratic codes with fewer images, so the capture time remains small.

3.3 Unsynchronized camera-projector system

As illustrated in Fig. 3.1, a projector plays a continuous video loop of a number of patterns (typically 30 or 60) at a fast frame rate, typically the same as the projector patterns. The patterns are random unstructured light patterns. The capture starts somewhere in the sequence without synchronization. The camera and the projector are assumed to have the same frame rate, so each projected pattern is seen exactly once, possibly mixed with another one, by the camera. In practice, it is important to give enough time to the camera to adjust itself, so we skip a number of images at the start of the capture. Also, when scanning deformable or moving objects such as faces, capturing more images can be helpful since it

allows us to choose the optimal sequence where the person moves the least. As we know, it is very hard for a human to remain still for more than two seconds. This is why we focus on scans that require two seconds or less.

Since the start of the camera capture is unsynchronized, it is essential to establish which pattern is seen first. Since we capture the same number of images as there are projected patterns, all the algorithm has to do is search across all captured images for the first pattern. However, it is impossible to recognize such a pattern, so another approach must be used.

3.3.1 Finding the first image of the sequence

As mentioned previously, the camera can start capturing at any time while observing a continuous video loop of patterns. The first step of our method is thus to find which image in the captured sequence corresponds to the first pattern. For this section, we assume that the images are not a blend of partially exposed consecutive patterns. Even if this is not true in practice, partial multiple exposures always feature a "dominating" pattern, which is enough for finding the first image.

First, we compute a binary code for the projector pattern sequence and the captured image sequence in their original order, both from their start. For speed purposes, the code is linear, not quadratic. Then, we compute a first correspondence between the projector and the camera, by running a small number of iterations of LSH (typically 6). The intent is not to obtain an actual match but just to detect if we are matching completely unrelated images or not. We calculate the sum matching costs after a few iterations of LSH. After that, we shift the camera codes to the left, effectively changing which image is considered the first, while the projector codes remain unchanged. We then proceed to match these codes again and keep the matching costs. After n shifts, we have tested all possible matches. All matches should have a similar high cost, and a single match will be lower, indicating we have found the first image. Once the minimum is found, we reorder the initial camera sequence according to its position found and the sequences are effectively aligned.

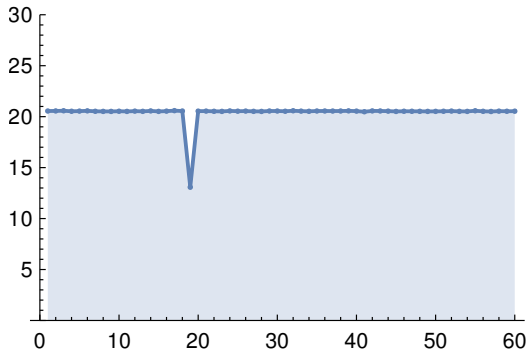


Figure 3.3: Matching costs for finding the first image. The average of the erroneous bits decreases from 22 to 13 on a 60 bits code. The x and y axes represent the pattern number and the number of erroneous bits, respectively.

For this step, we use linear code because the gap between the sum of the first image and the other sums is large, even after the first iterations of LSH, as shown in Fig. 3.3. The matches obtained are affected by the partial exposure of the camera, but the gap between the sums is not really affected. That's why we have assumed above that our images are perfectly exposed. In the next section, we will introduce partial exposure that affects our captured images.

3.3.2 Partial exposure of consecutive patterns

When capturing asynchronously, as illustrated in Fig. 3.1, it is entirely possible that the projected pattern will change during the exposure of a single image. The resulting image will feature a multiple exposure, a mixture, of two consecutive projected patterns. When this happens, matching will become harder, or even impossible. This problem is even worse for rolling shutter cameras, since the exposure start time varies from the top to the bottom of the camera image. In these conditions, matching with the original projected patterns will yield a partial match, at best, as illustrated in the top left of Fig. 3.4.

In the presence of partial exposures, the *finding first image* step previously presented will still work. The *longest* exposure of the two partial exposures is the reference and the

other partial exposure acts as noise. The matching can proceed but is affected by the noise. However, to obtain the best possible match, we will estimate the mixture of the partial exposure and use both exposures as the reference to get a better match. Once the mixture is known, "the partial exposure" is not noise any more.

Unstructured codes, contrary to structured codes, allow total freedom in the selection of the projected patterns. This property will now become very useful. Instead of considering an image as a partial exposure of two consecutive patterns, we will consider that the image is a full exposure of a new reference pattern which is built from a mixture of two consecutive patterns. Fig. 3.4 illustrates the matching results obtained for various mixing of consecutive patterns.

The method for estimating this "consecutive pattern mixture" is described in the following section.

3.3.3 Mixture of captured images

As defined above, the lack of synchronization can induce a multiple exposure between two consecutive patterns, as illustrated in Fig. 3.5. The first and second cases (top and middle) are typical variations of the exposure between the camera and the projector. This variation generates the mixture of two consecutive patterns. The mixture represents a percentage (50:50 or 75:25) of each projected pattern. The last example is a rare case (bottom) where the frame rates of the camera and the projector are synchronized, so the camera capture a perfect exposure of only one pattern.

If the frame rate durations of the camera and the projector are not matched, then the mixture will change in time. When such a mismatch is present, various temporal mixing strategies can be used, but at an increased computation cost. Considering that frame rates are usually stable in time, it is easy to devise a way to measure those frame rates beforehand. Then, it is much easier to take these different frame rates into account in the matching method. In practice, when the camera and projector frame rates are similar enough (i.e. 60

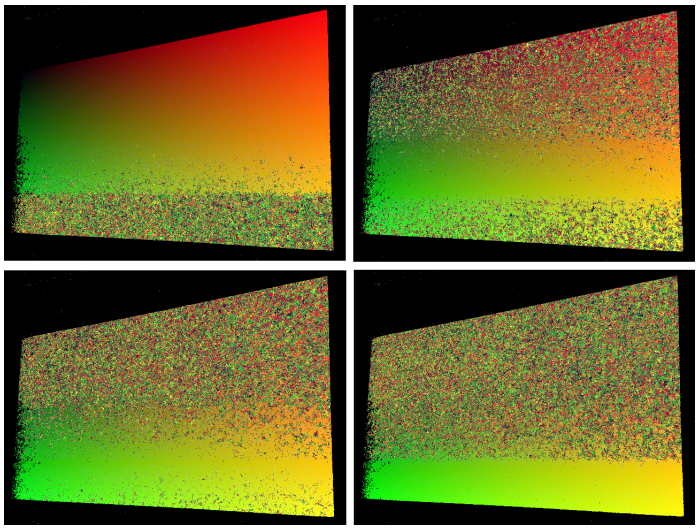


Figure 3.4: Matching obtained for various partial exposures of two consecutive patterns. In each matching, x and y coordinates are represented as red and green, respectively. The mixtures illustrated are 0, 0.5, 0.6, and 1.0

fps and 59.94 fps), the variation in mixture over a few seconds is negligible. This is our assumption in this paper's experiments.

For a global shutter (i.e. progressive) camera, we expect that a single mix value will be used across the whole camera image. We could then *search* for the best mix value and then proceed to match with that value. On the other hand, for a rolling shutter camera, the mix changes vertically across the camera image.

In our method, we will allow one mix value per pixel, so rolling shutter as well as global shutter camera are supported. However, we assume that the mix value does not change in time during the capture. This implies that the frame rates of the camera and projector are well matched.

First, we must find the two consecutive patterns that are mixed in the captured images. The image $\mathrm{I_i}$ can be mixed with the previous pattern $(i-1)$ or the next one $(i+1)$, so we must evaluate both cases. We calculate the error of the two cases over a few iterations of LSH. The smallest error indicates which of the two cases (let's call them $d=-1$ and

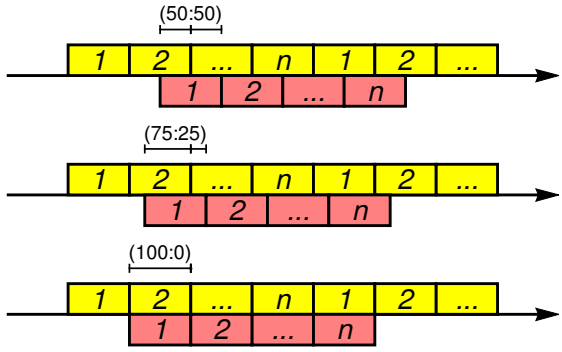


Figure 3.5: Different mix values (in percentage) that represent three cases of a partial exposure (50%, 75% and 100%) of two consecutive patterns. The last image represents a rare case of a synchronized capture without mixture (bottom).

$d = +1$) is the one contributing to the mixture of images.

From now on, we will explore a number of possible mixtures of patterns i and $i+d$. For each tested mixture μ, we will generate a new set of reference images $I'(i) = \mu I(i) + (1-\mu)I(i+d)$. The camera image sequence will be matched to these reference patterns and any resulting good match will be kept. In practice, we test the set of mixtures from 0 to 1, with a step of 0.1. For each pixel, its final match will be the one with minimal cost over all these mixtures. This not only provides robustness to the distribution of mixtures in the image, but provides a simple estimation of that mixture, as illustrated in Fig. 3.9.

For this part, matching relies on quadratic codes, because we want the maximum amount of information for each pixel to increase the quality of the match. For the same reason, we also increase the number of iterations of LSH (typically 80 iterations). In the bottom of Fig. 3.6, the matching result is provided as a Lookup Table that we obtained after finding the first image and the mix. As an illustration of the effectiveness of this approach, we illustrated in Fig. 3.4 separate results for different mixtures and a rolling shutter camera. It can be clearly seen that each mixture yields an associated correct match, and combining these matches will result in a correct solution. Contrary to [35], the optimal mixture is obtained without explicitly solving an internal camera imaging model.

3.4 Experiments

In this section, we present the experimental evaluation of our proposed method on real scenes. Since all the current methods that need to synchronize the projector-camera system use structured light, we cannot compare them directly with our method because we use unstructured light patterns. So we made synchronized and unsynchronized matches to assess the accuracy and quality of the unsynchronized method itself, not the LSH algorithm. We provide results for a varying number of patterns, different frame rates, for two different scenes. The first set of results uses quadratic code and the second set uses linear code. The goal of our method is to make possible a 3D scanner with the simplest camera-projector hardware possible. For the realization of all our experiments, we used non-professional hardware. The projector was an Aaxa HD Pico projector used at its native resolution of 1280x720 for all scans, both at 30 and 60 fps. Two cameras were used, at different frame rates. A Logitech C920 webcam was used at 1920x1080 resolution at 30 fps (its maximum rate) and a GoPro HERO3+ was used at a resolution of 1920x1080 for scans at 60 fps. Both cameras are rolling shutter with automatic brightness adjustment, the so-called *auto gain*. This type of camera can cause problems during a scan because it tries to adapt to lighting changes throughout the capture. Fortunately, unstructured light patterns are known and appreciated for their constant average intensity, so they are well suited for this kind of camera. Finally, we encountered the problem of flickering during the experiments. This flicker depends on the light source observed. Most ordinary projectors will create RGB color images by presenting the RGB planes one color after another. When the exposure time is short, a camera sees these individual color planes as a flicker (the *color wheel* effect). It is thus important to ensure that the camera exposure time is as long as the frame rate will allow ($\frac{1}{60}$sec is perfect for most projector).

In our experiments, we projected continuous video loops of 30, 60 or 120 patterns. For a first set of experiments, they were projected at 30 fps on two scenes ("plane" and "vase") and observed at 30 fps by the Logitech camera. For a second set of experiments,

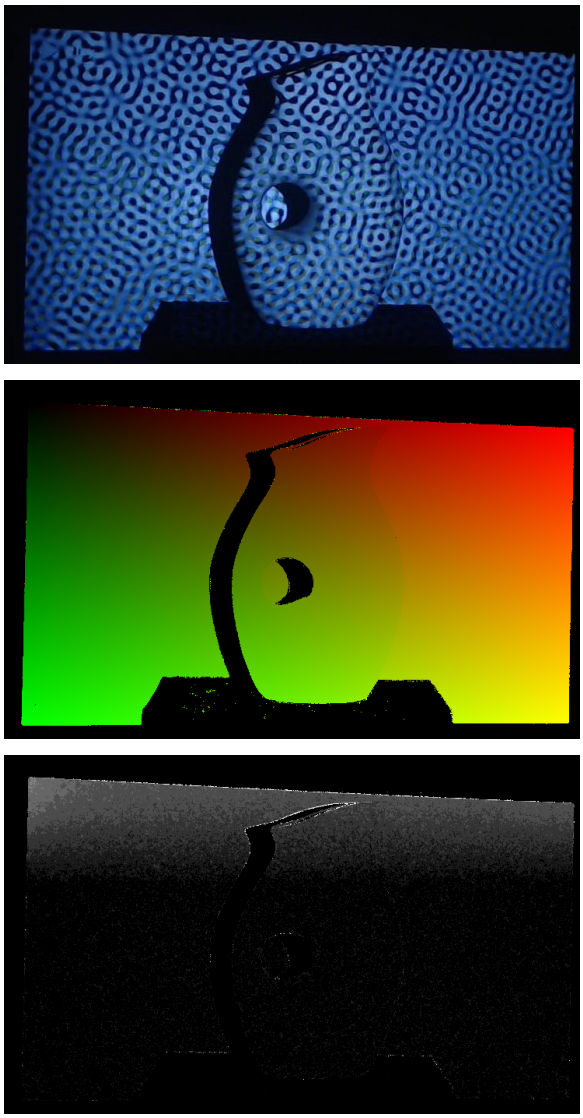


Figure 3.6: Projected patterns on a typical scene, observed by the camera (top), camera-projector Lookup Table (middle) where red and green represent x, y positions, and computed mix of successive patterns (bottom). The mix value changes from top to bottom, indicating a rolling shutter camera.

fps	scene	N	std	std ref	loss
30	plane	30	0.65	0.22	0.43
		60	0.49	0.29	0.20
		120	0.49	0.04	0.45
	vase	30	0.86	0.23	0.63
		60	0.66	0.28	0.38
		120	0.35	0.04	0.31
60	plane	30	0.76	0.34	0.42
		60	0.55	0.17	0.38
		120	0.41	0.09	0.32
	vase	30	0.77	0.34	0.43
		60	0.55	0.15	0.40
		120	0.81	0.08	0.73

Table 3.1: The standard deviation of displacements (in pixels) in x, y obtained with a different number of patterns and a different fps onto two scenes. Std represents the standard deviation of differences between synchronized and unsynchronized. Std ref represents the standard deviation between two synchronized scans. Loss is the difference between the two standard deviations.

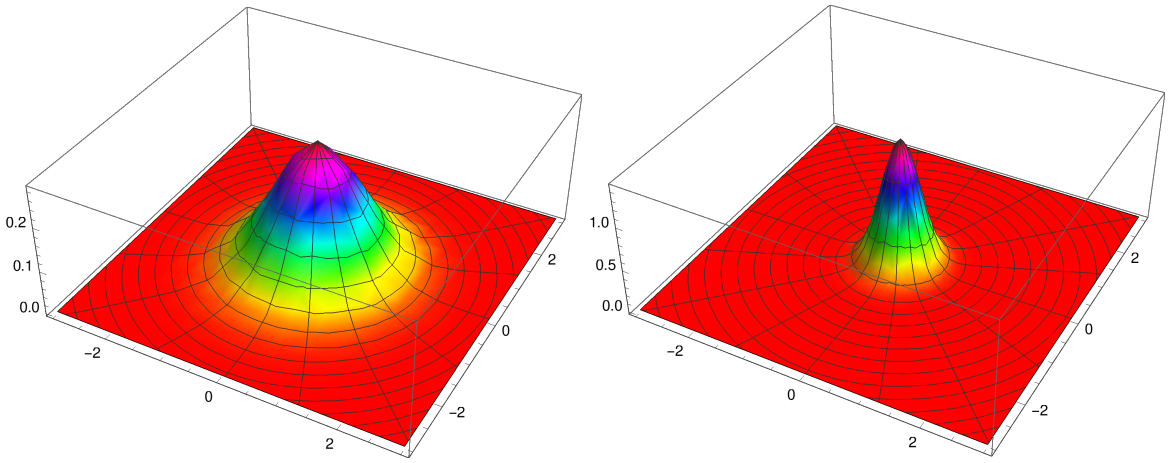


Figure 3.7: Average displacements (in pixels) between unsynchronized matching and synchronized matching (left), and for running twice synchronized matching on the same images (right). The maximum precision on the right is limited by the number of LSH iterations.

they were projected at 60 fps and observed at 60 fps by the GoPro camera. A final set of "synchronized" experiments was also performed with projection at 5 fps, observed from both cameras, to obtain reference matches for comparison.

The results are provided in Table 3.1.

In order to evaluate only the performance of the synchronization aspect of our method, we compared unsynchronized matchings with synchronized matchings. This ensures that we do not compare the structured light method itself, but only the impact of synchronization. The column "std" indicates the standard deviation in pixels observed between unsynchronized and synchronized results, in pixels, in the camera reference frame at 1920x1080 resolution. We observe deviations of fewer than one pixel in all cases, which illustrates how well the method works.

However, since the matching algorithm is probabilistic (LSH), its solution varies. The column "std ref" measures this variation by comparing two synchronized matching obtained on the same captured images. These values, which are generally very small, con-

stitute the precision limit of the method (at the selected number of LSH iterations). This implies that the true *loss of precision* resulting from removing synchronization is in column "loss", expressed as the difference between the deviations of the two columns "std" and "std ref".

In analyzing those results, one must consider that the matchings are all computed with integer pixel positions. This artificially increases the standard deviations when it is lower than one pixel. A new set of experiments should be devised eventually to use sub-pixel matching, since the observed accuracy seems to justify it.

Overall, the results are very good and illustrate that the loss of precision for our method is minimal.

Typical distributions of matching errors, measured as the number of different bits between codes, are illustrated in Fig. 3.7. The curve on the left corresponds to the differences of a synchronized scan and an unsynchronized one. The curve on the right corresponds to comparing two synchronized scans run on the same captured images. We observe that the distributions are approximately normal with a larger standard deviation for unsynchronized matching, as expected.

For the last experiments, we used the same sets of patterns and captured images (60 and 120 patterns projected onto a "plane") to generate the Lookup Table using a linear code. Fig. 3.8 illustrates the errors distribution for a synchronized and an unsynchronized matching. They are very similar, indicating that the lack of synchronization does not make the matching more difficult.

Notice that the synchronized distributions are binomial, with n trials corresponding to the linear code-length in bits (here 60 and 120) and a probability p related to the capture conditions. The unsynchronized distributions (especially with 120 bits) are modeled as a mixture of two binomials, one being the same as the synchronized case, and one with a slightly larger p which applies for more difficult mixed exposure cases. Overall, the variation between the two curves in Fig. 3.8, synchronized in red and unsynchronized in green, are small.

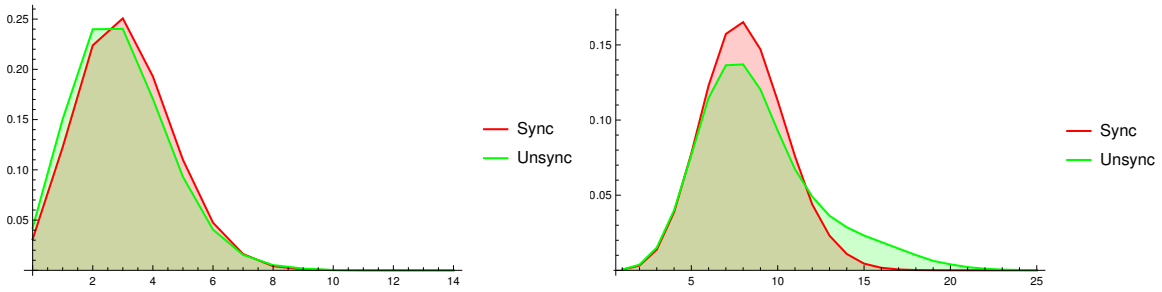


Figure 3.8: Curves representing the probability density of matching errors, measured in bits, for unsynchronized matching (green) and synchronized matching (red). These experiments were generated with 60 (top) and 120 (bottom) patterns and matched with a linear code.

One of the goals of our method is to make scanning faces easier. Scanning in less than two seconds makes this much easier. Fig. 3.9 shows the different mix values obtained while scanning a face, illustrating that the method works well for these kind of applications. Adding calibration and sub-pixel accuracy to the matching process, we expect to obtain excellent 3d models of faces.

3.5 Conclusion

In this article, we presented a novel unsynchronized active scan method based on an unstructured light framework. Tested at the limit of current low cost cameras and projectors, at 30 and 60 fps, we obtained results that are extremely close to what can be obtained with synchronization. We expect this method to make scanning in some difficult situations much more feasible, such as scanning human faces or large objects where the camera is too far from the projector to be easily synchronized. In the future, it should be straightforward to add sub-pixel accuracy to the matching algorithm, make it faster, parallel and fully test its performance on full 3D reconstructions. Also, using a custom fast projector and industrial camera, it should be possible to achieve very high capture rates, which could open new

applications.

Figure 3.9: Selected best mix values for a rolling shutter camera. The mix values range from 0 to 1, corresponding to increasing levels of partial exposures between consecutive patterns. The image on the left represents a scan seen by the camera and the image on the right by the projector.

Chapitre 4

RECONSTRUCTION 3D

La reconstruction 3D consiste à obtenir un modèle 3D à partir des correspondances denses encodées dans les *LUTs*. Nous récupérons la position de chaque pixel dans l'une des *LUT* caméra-projecteur soit projecteur-caméra. Ensuite, il suffit de calculer les correspondances de ces pixels. Pour obtenir le modèle 3D, il faut finalement trianguler les pixels et leurs correspondances. Toutefois, il nous faut récupérer la position de la caméra et du projecteur dans le monde et ainsi que leurs paramètres internes afin de pouvoir effectuer la triangulation. Retrouver ces paramètres dits internes et externes est ce que nous appelons le calibrage. Ce chapitre détaille le processus du calibrage de la caméra et du projecteur. Ensuite, le concept de la triangulation sera décrit, accompagné d'exemples de modèles 3D.

4.1 Calibrage

Comme mentionné ci-haut, il est nécessaire de calibrer le système caméra-projecteur pour obtenir un modèle 3D. Il faut donc estimer la pose de la caméra et du projecteur l'un par rapport à l'autre dans le monde. Ainsi, il faut réaliser le calibrage de chaque dispositif qui va être utilisé lors de la triangulation. Pour alléger le texte dans cette partie, nous allons détailler le processus du calibrage de la caméra, mais le même processus s'applique au projecteur. Ensuite, nous allons expliquer plus spécifiquement les deux méthodes utilisées pour calibrer notre caméra et notre projecteur.

Le calibrage est la relation entre le monde et l'image formée par la caméra. Il existe une transformation entre des points 3D dans le monde et les points équivalents en 2D dans l'image. Cette transformation peut être divisée en deux; le passage des points 3D du monde vers le modèle de la caméra et ensuite vers les points 2D de l'image. Le premier passage

est le fait d'estimer la position c et la rotation R de la caméra dans le monde par rapport au système de coordonnées du monde. Ceci est appelé paramètres extrinsèques de la caméra. Le deuxième passage est celui de la transformation des points du modèle de la caméra vers leur équivalent en pixels dans l'image. Celle-ci est appelée paramètres intrinsèques, et ceux-ci sont représentés par la matrice K. Ces paramètres ne changent pas si la caméra est déplacée, car ce sont les caractéristiques de la caméra elle-même. Le modèle de caméra obtenu par le calibrage définit entièrement le processus de formation de l'image.

Le modèle de caméra perspective est basé sur le modèle de caméra sténopée (*pinhole*). C'est une caméra avec une profondeur de champ infinie qui n'a donc pas de focus. Nous choisissons ce modèle, bien qu'il ne soit pas toujours parfaitement réaliste, parce qu'il simplifie le processus de calibrage. Le modèle de projection perspective à résoudre est donc $P = KR[I|c]$.

$$P = \underbrace{\begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{K}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\text{I}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{R}} \underbrace{\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{\text{T}}$$

$$\underbrace{\underbrace{\text{paramètres internes}}_{\text{K}} \underbrace{\qquad\qquad}_{\text{I}}}_{\text{projection perspective}} \underbrace{\qquad\qquad\qquad\qquad}_{\text{paramètres externes}}$$

Dans cette équation, f, la distance focale, dépend de l'angle de vue de la caméra et (cx, cy) représentent l'intersection de l'axe optique avec l'image. Habituellement, ce point se situe au centre de l'image soit $(\frac{W}{2}, \frac{H}{2})$.

Il existe plusieurs méthodes pour effectuer l'évaluation des paramètres de calibrage; le calibrage avec un objet 3D connu, le calibrage par rotation pure avec une caméra qui tourne autour de son centre optique et le calibrage planaire à partir de plusieurs vues d'un objet planaire.

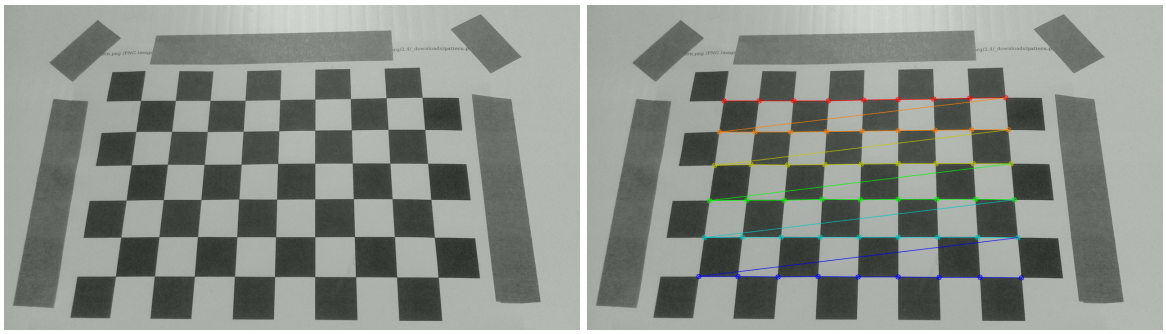


Figure 4.1: Un damier (à gauche) à l'origine du monde du point de vue de la caméra. La figure (à droite) représente des points saillants (les coins du damier) sélectionnées par *OpenCV* sur un damier.

4.1.1 Calibrage de la caméra

Afin d'effectuer le calibrage de la caméra dans ce projet, nous avons utilisé une caméra perspective ordinaire. La méthode de calibrage choisie est le calibrage planaire. La méthode consiste à prendre en photos un objet planaire de plusieurs vues différentes. Généralement, l'objet planaire de référence choisi est un damier (Fig. 4.1 à droite), car il présente une grille de points aux distances connues dans le monde. Pour chaque vue d'un damier, les points saillants (les coins des cases) sont récupérés pour leur attribuer les données. En d'autres termes, nous mettons en correspondance les coordonnées dans l'image en pixels et les coordonnées des points du damier dans le monde (Fig. 4.1 à gauche). Le nombre et la taille des cases sont connus. Le damier est défini à l'origine du monde avec son plan situé à une profondeur $z = 0$. La pose et la rotation de la caméra (les paramètres extrinsèques) sont récupérées par rapport au plan pour chaque vue. Inversement, les paramètres intrinsèques sont calculés à partir de toutes les vues grâce aux contraintes sur les homographies et ne changent pas lorsque la caméra est déplacée. Chaque damier comporte une homographie H.

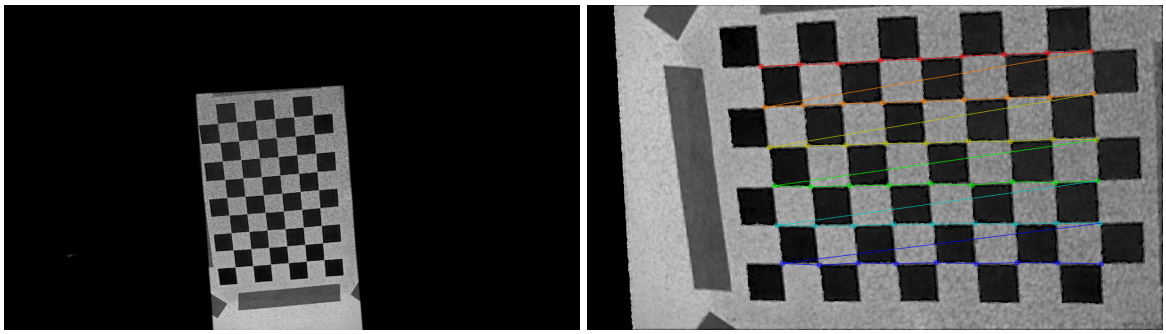


Figure 4.2: Le damier (à gauche) représente l'image moyenne d'une projection de patrons à lumière non structurée du point de vue de la caméra. Le damier (à droite) représente l'image obtenue du point de vue du projecteur représentant des points saillants. Le projecteur possède une rotation de 90° par rapport à la caméra.

$$H = K\,R \begin{bmatrix} 1 & 0 & -c_x \\ 0 & 1 & -c_y \\ 0 & 0 & -c_z \end{bmatrix}$$

Dans notre cas, la librairie *OpenCV* [1] est utilisée pour calibrer la caméra.

4.1.2 Calibrage du projecteur

Pour le calibrage du projecteur, nous utilisons la même méthode de calibrage planaire que celle employée pour la caméra. En effet, un projecteur peut être considéré mathématiquement comme une caméra. Puisque le projecteur ne peut pas "observer" une scène, la scène sera observée avec une caméra dont l'image sera ensuite transformée pour fournir le point de vue du projecteur. L'idée est de récupérer les damiers du point de vue du projecteur et ainsi calculer les paramètres intrinsèques et extrinsèques à l'aide de *OpenCV* [1], tel qu'expliqué ci-haut. Pour effectuer ceci, une propriété de la méthode lumière non structurée est utilisée soit la bidirectionnalité des correspondances entre le projecteur et la caméra. Grâce à cette propriété, l'image capturée par la caméra peut être transformée vers

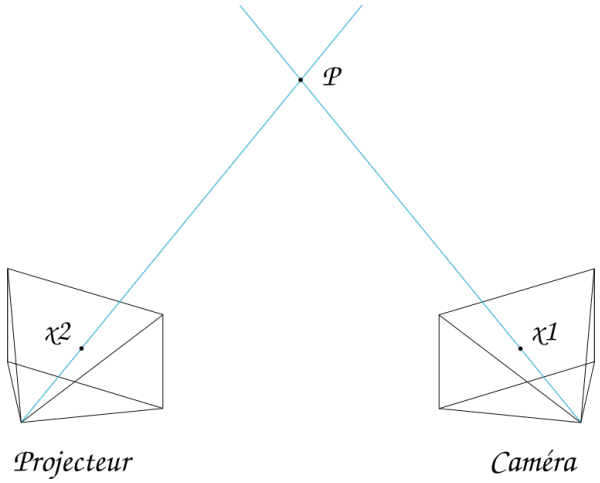


Figure 4.3: Processus de triangulation. $\{x_1, x_2\}$ représentent les centres optiques de la caméra et du projecteur, respectivement. *P* représente le point 3D triangulé.

le point de vue du projecteur. En d'autres termes, il faut appliquer les correspondances du projecteur sur l'image de la caméra, et ce à l'aide de l'image moyenne de la caméra (Fig. 4.2 à droite).

Les mises en correspondance sont réalisées à partir de plusieurs vues différentes d'un damier. Ensuite, l'image du damier est composée du point de vue du projecteur, comme le montre la Fig. 4.2 (à gauche). Finalement, plusieurs vues différentes du damier "capturées" par le projecteur sont obtenues. Ainsi, il est possible de calibrer le projecteur comme une caméra en utilisant un calibrage planaire.

4.2 Triangulation

La triangulation réfère au processus qui détermine la profondeur d'un point 3D à partir de deux points dans des images prises de différents points de vue, comme le montre la Fig. 4.3. Tout d'abord, il faut récupérer la position des pixels dans les *LUTs* (caméra vers projecteur ou projecteur vers caméra) et leurs correspondances pour ensuite générer un nuage de point en 3D. La triangulation est appelée aussi reconstruction 3D [23].

Il est important de calibrer le projecteur et la caméra pour résoudre ce problème. Une fois que la position d'une caméra est connue dans le monde, il est possible de calculer le rayon dans le monde correspondant à la trajectoire de la lumière qui se projette sur un point donné de l'image, en passant par le centre optique. Pour chaque point de vue, nous calculons les droites dans le monde qui passe par les points correspondants des deux images. L'intersection de ces deux droites détermine la position du point 3D et, par conséquent, sa profondeur dans le monde. Après avoir triangulé plusieurs paires de points correspondants, un nuage de points en 3D est construit. Enfin, des polygones sont calculés sur le nuage de points pour former des triangles et ainsi constituer une surface 3D. Pour réaliser ce processus de triangulation, la librairie *OpenCV* [5] est utilisé.

Chapitre 5

(ARTICLE) SUBPIXEL UNSYNCHRONIZED UNSTRUCTURED LIGHT

Ce chapitre présente l'article [15] en préparation pour publication tel qu'indiqué dans la bibliographie:

> C. El Asmi et S. Roy, *Subpixel Unsynchronized Unstructured Light*, À soumettre.

Dans cet article, nous améliorons la méthode à lumière non synchronisée non structurée présentée au chapitre 3 en augmentant la précision des correspondances entre la caméra et le projecteur. Une correspondance sous-pixel permet d'améliorer la qualité des reconstructions 3D obtenues en les rendant plus lisses. Généralement, on considère qu'un pixel de la caméra correspond exactement à un pixel du projecteur. Il est possible que la vraie correspondance d'un pixel de la caméra se situe réellement entre deux pixels voisins du projecteur, et vice versa. En calculant le sous-pixel, il devient possible de retrouver des correspondances de plus haute qualité, avec des déplacements qui ne sont pas entiers. La précision sous-pixel peut varier selon plusieurs facteurs tels que le ratio de pixels ou la fréquence des patrons, mais elle augmente toujours la qualité de la reconstruction 3D obtenue.

Le ratio de pixels correspond au nombre de pixels qui sont mis en correspondances entre le projecteur et la caméra. En augmentant le ratio de pixels (ratio = 4 par exemple), un pixel de la caméra voit 4 pixels voisins du projecteur. Il est alors possible d'améliorer la correspondance projecteur-caméra avec du sous-pixel. À l'inverse, dans cette situation, la correspondance caméra-projecteur est déjà sous-pixel. En augmentant la fréquence des patrons, la précision du sous-pixel augmente. Or, si la fréquence devient très haute alors la

caméra voit majoritairement du gris et ainsi même le sous-pixel ne peut plus améliorer la qualité des correspondances.

Les résultats obtenus par méthode à lumière non synchronisée non structurée sous-pixel sont comparés à des résultats obtenues par méthode à lumière non synchronisée non structurée sans sous-pixel et par méthode Phase Shift. Un modèle 3D d'un visage reconstruit à l'aide de notre méthode permet de valider le but initial de ce mémoire.

L'article est présenté dans sa version originale.

Abstract

This paper proposes to add subpixel accuracy to the unsynchronized unstructured light method while achieving high-speed dense reconstruction without any camera-projector synchronization. This allows scanning faces which is notoriously difficult due to involuntary movements on the part of the model and the reduced possibilities of 3D scanner approaches such as laser scanners because of speed or eye protection. The unsynchronized unstructured light method achieves this with low-cost hardware and at a high capture and projection frame rate (up to 60 fps). The proposed approach proceeds by complementing a discrete binary coded match with a continuous interpolated code which is matched to subpixel precision. This subpixel matching can even correct for erroneous camera-projector correspondences. The obtained results show that highly accurate unfiltered 3D models can be reconstructed even in difficult capture conditions such as indirect illumination, scene discontinuities, or low hardware quality.

5.1 Introduction

The subpixel correspondence is very important in 3D reconstruction as it enables a smooth and dense 3D model. Generally, active reconstruction produces a correspondence where one camera pixel corresponds to one particular projector pixel. On the other hand, by achieving a subpixel correspondence, the accuracy is greatly improved as it improves the matches and enables pixels to be matched to a fractional part of another pixel, as illustrated in Fig. 5.1.

There are multiple active reconstruction methods that can provide a subpixel correspondence. These methods are divided into two broad categories which can further be split into multiple methods. These methods are referred to as the structured light method and the unstructured light method. The first category consists of projecting several structured light patterns and directly encoding the position of the projector pixel. In this category, the first method is the *Gray Code* [26] and the patterns are composed of white and black stripes at

different frequencies. A second method is the *Phase Shift* [47] where sinusoidal patterns, composed of the same sine shifted several times at different frequencies, are projected. These methods exhibit many difficulties in scene discontinuities and they are not robust to indirect illumination which in turn leads to multiple matching errors. Several other approaches have tried to improve the *Phase Shift* [11, 20, 18]. These methods will be detailed in the next section. The second category, unlike the previous one, consists in encoding the position of the projector and the camera in a *LookUp-Table* (LUT) [32, 14, 52, 13]. The unstructured light method provides bidirectional matching (from camera to projector and from projector to camera). In [12], they improved the patterns by generating sines in random directions in the frequency domain. Additionally, these patterns don't feature large black and white regions. For this reason, this method is very robust to indirect illumination and scene discontinuities.

The methods presented above must synchronize their projectors and their cameras. Without synchronization, the camera sees mixed projected patterns which results in wrong correspondences. To obtain a correspondence from patterns projected in time, the camera must see each projected pattern by the projector only once. There are two types of synchronization; hardware synchronization [49, 54, 55, 41, 33, 51] and software synchronization [24, 37, 31, 28]. The first type requires expensive and experimental equipment. It consists in synchronizing the projector and the camera using a *triggering circuit* [33, 51]. This type of synchronization allows the capture of image sequences at very high frame rate (up to *3000 fps* [49]). The second type does not require any experimental material. It is a structured light scan at very low frame rate (usually less than *5 fps*). Unfortunately, this method, with its low frame rate, requires a large amount of time for the camera to fully capture the projected patterns exactly once.

Other methods have performed unsynchronized coded light scans [58, 42, 35]. The difficulties of the unsynchronized capture reside in finding the first image in the captured sequence and in finding the mixture between two consecutive patterns partially seen by the camera as a single image. Indeed, during the unsynchronized capture at very high frame

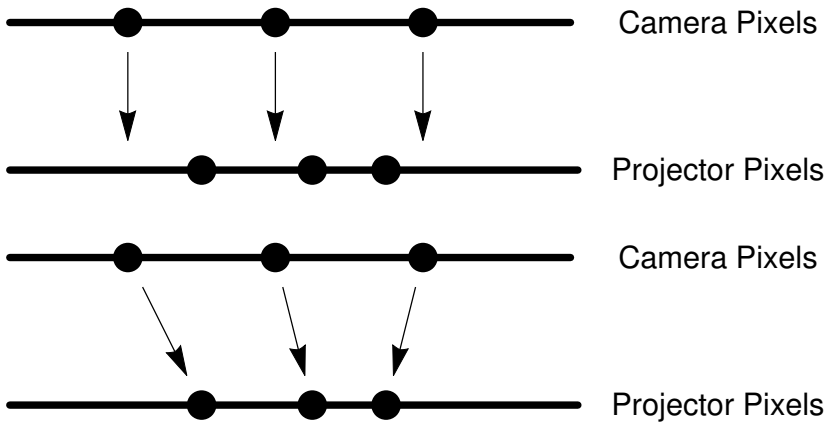


Figure 5.1: Obtained pixel correspondence between the camera and the projector (top) and a correspondence with subpixel accuracy between the camera and the projector (bottom) which means a pixel can be matched to a fractional part of another pixel.

rate, the camera sees a mixture of two consecutive patterns. It then becomes impossible to get a correspondence between the camera and the projector.

The first method [35] consists in projecting structured light patterns at a high frame rate without synchronization between the projector and the camera. The authors project a looping video of structured light patterns. In order to detect the first image in the captured sequence, they project an easily identifiable sequence of entirely black and entirely white patterns at the beginning of the sequence. They then generate an image formation model of the camera in order to find the synchronization parameters and to recover the patterns corresponding to the *Gray Code*. This method requires complex and very long computations in order to solve the equation systems of the image model. In addition, it is not robust to indirect illumination and scene discontinuities due to the use of *Gray Code*.

Alternative methods [16] solved the synchronization problem by projecting a looping video of unstructured light patterns at a high frame rate (*30 to 60 fps*). The camera starts capturing at any time. Thus, it is necessary to find the first image of the captured sequence. They do so by making several correspondences between the captured sequence and the ref-

erence sequence which is shifted by one pattern at each correspondence. The first image in the captured sequence is found using the best correspondence after calculating the matching costs. They then find the mixture between the two consecutive patterns by mixing them. The unstructured light patterns are generated randomly so mixing them gives a new random pattern. This method is very fast and simple. It can scan in less than two seconds at *30* or *60 fps*. However, this method does not achieve a correspondence with a high subpixel accuracy. In this paper, we describe a new technique to improve the unsynchronized unstructured light method by matching with a high precision subpixel.

5.2 Previous work

There are several active methods that achieve a high precision subpixel correspondence. In articles [46, 45], a survey on structured light methods is presented. In general, methods that achieve subpixel precision are based on sinusoidal patterns [53, 56]. The patterns are composed of multiple sines each shifted by a different amount in a given direction and with different frequencies. The sines vary from a very low frequency to a very high frequency. Thus, each camera pixel encodes the projector position directly by a unique phase. This method achieves a dense reconstruction with a high subpixel accuracy through the different gray intensities. However, this method requires photometric calibration because the phase is recovered from the pixel intensities. Furthermore, it is not robust to the indirect illumination which is caused by the low frequency patterns.

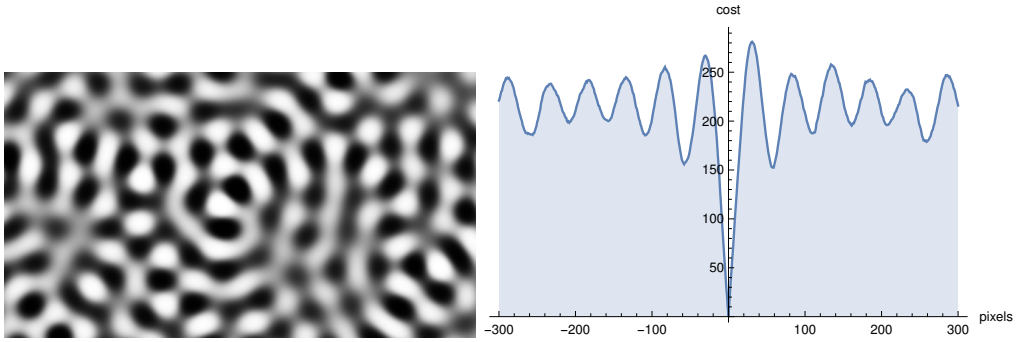
In [11], they improved the projected patterns by modulating a high frequency signal, so that they are robust to indirect illumination and achieve a high subpixel accuracy. *Modulated Phase Shift* patterns are composed of modulated sines in both directions (two-dimensional patterns) at a very high frequency. Unfortunately, this method requires a very high number of patterns. In [18]'s method, they reduced the number of patterns by multiplexing the modulated patterns together. These three methods require what is called the *phase unwrapping* because of the periodic nature of patterns [25, 36]. Indeed, we must be

able to differentiate between the different phases of each period. *Micro Phase Shift* method [20] resolves the problem of *phase unwrapping* by projecting only a high frequency patterns. Alternative methods have used the *Gray Code* [19] to achieve a subpixel reconstruction. *Line Shifting* [19] evaluates the subpixel only in the bit transitions (0 to 1 or 1 to 0). However, these alternative methods result in a sparse reconstruction.

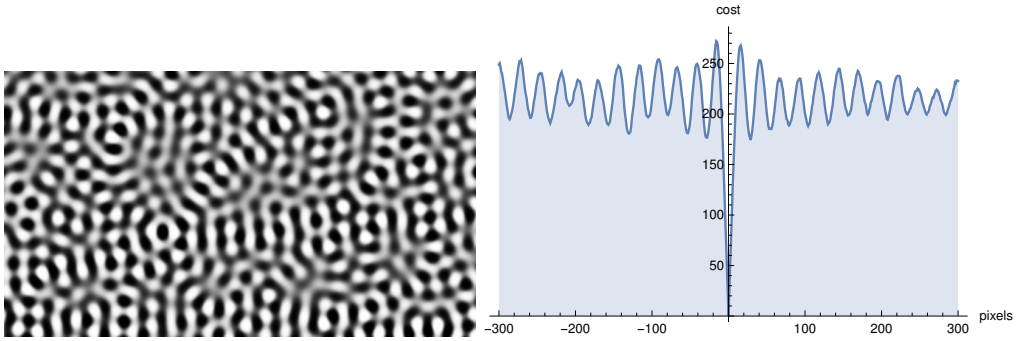
In [34], they use the unstructured light method to achieve the subpixel accuracy. This method is very robust to indirect illumination and scene discontinuities through their gray level band-pass white noise patterns. They project a lower number of patterns than the method in [12]. They also improved their technique to generate the codewords [46]. By comparing two neighboring codewords, they determine the region where the subpixel is located. They then divide it into four bins by interpolating between the four pixels that define this region. They additionally make a hierarchical vote to choose the right bin and further divide it into another four bins. This operation is repeated recursively several times until they obtain the desired amount of subpixel precision. This method requires a huge calculation time because of the recursion and the hierarchical vote. In this paper, the unsynchronized unstructured light method [16] is improved by accomplishing a high subpixel accuracy. A simple and fast technique to determine the subpixel position is presented in Sec. 5.4.

5.3 Relevant subpixel information

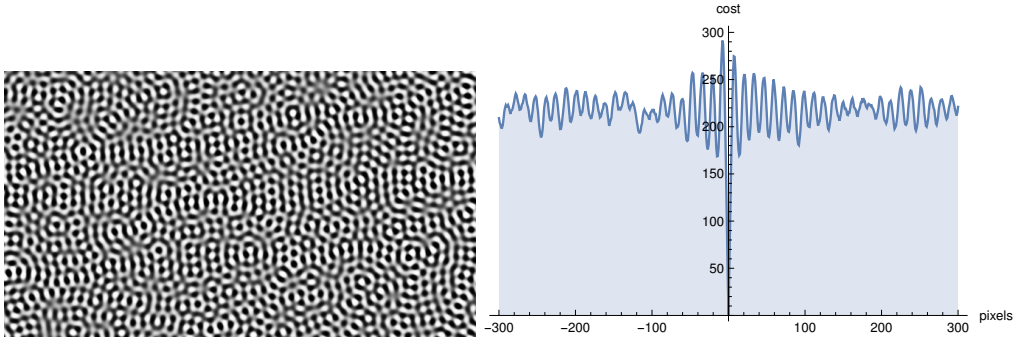
In establishing pixel correspondence with unstructured light patterns, several parameters have an impact on subpixel accuracy. Amongst these parameters, there is the pattern frequency and the pixel ratio as well as the code-length (linear and quadratic code). Modulating these parameters allow the subpixel accuracy to either improve or degrade.



(a)



(b)



(c)

Figure 5.2: Unstructured light patterns at various spatial frequencies and their cost functions representing the cost of the difference between two neighboring pixels (here, a neighborhood of 300 pixels). The frequency represents the oscillation number of each sine per pattern. Notice that when the frequency increases, the curve is more pronounced. Fig. (a) shows a pattern frequency equal to 25, (b) shows a pattern frequency equal to 50 and (c) shows a pattern frequency equal to 100.

5.3.1 Pattern frequency

The unstructured light pattern frequency is the oscillation number of one sine per pattern and is the main property of the unstructured light patterns. Increasing the pattern frequency reduces the impact of indirect illumination and improves matches. Using a very low frequency results in a high correlation between neighboring pixels as they become too similar to match effectively. The subpixel accuracy increases when the frequency is high because the curves of the cost functions are more pronounced and smooth. Fig. 5.2 shows three patterns with different frequencies and their associated cost function curves. As shown in the figure, the curve becomes more pronounced and precise as the frequency increases. However, using a very high frequency brings about several matching errors because the camera might not be able to distinguish the black and white bands.

5.3.2 Pixel ratio

The pixel ratio represents the number of pixels seen by a single camera pixel in the projector pattern, and vice versa. The optimal case is for the pixel ratio to be near 1. Indeed, a single pixel of the camera corresponds to only one pixel in the projector. For the current experiments, the pixel ratio is near 2 because the camera sees a mixture of four neighboring pixels in the projector (two pixels per axis). The subpixel accuracy decreases as the pixel ratio increases. To illustrate, consider an example of a pixel ratio near 2. If the camera “sees” four neighboring projector pixels then the correspondence from projector to camera already has a subpixel accuracy of a half pixel per axis. This is because the projector pixels have more information and they are more accurate. As illustrated in Fig. 5.3, we are already "inside" the camera’s pixels. Thus, the pixel ratio is very important in the determination of the subpixel matching, as it can increase or decrease its precision.

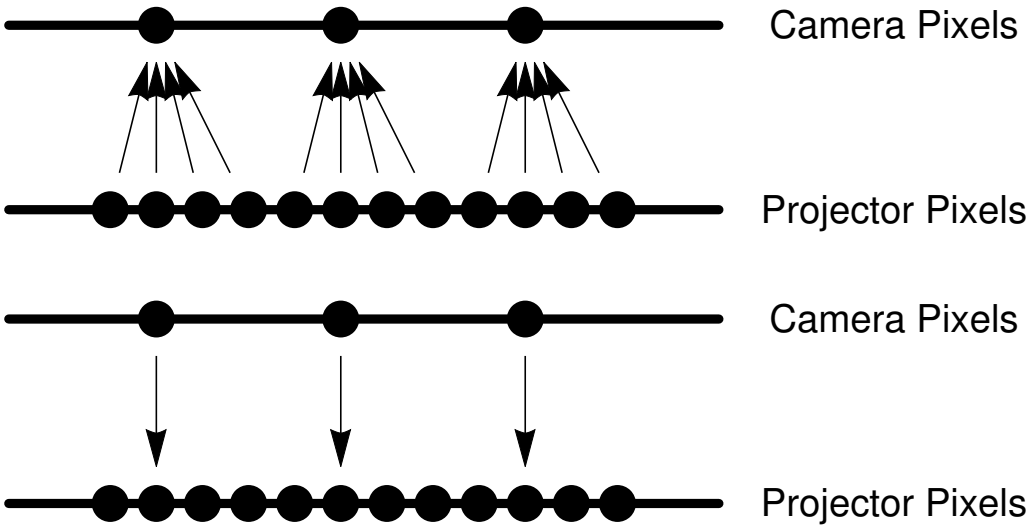


Figure 5.3: Illustration of pixel ratio where four projector pixels see the same camera pixel (top) and only one camera pixel sees a mixture of four projector pixels (bottom). One can say that the projector-camera correspondences is already subpixel whereas the inverse camera-projector correspondences isn't and it can be improved with a subpixel accuracy.

5.3.3 Linear and quadratic code

Pixel correspondences between camera and projector are established by using LSH algorithm (Locality Sensitive Hashing) [7]. LSH is used in searching for nearest neighbors in very high-dimensional spaces. Because of its inherently random nature, it is necessary to run several LSH iterations. At each iteration, it generates different match proposals and keeps only the best ones based on the difference of bits in the codes. While trying to recover subpixel accuracy, codes from neighboring pixels will be compared. These codes tend to be very similar, so we rely on quadratic code instead of linear code to get enough information.

As described in [16], a linear code with a small number of LSH iterations is used to find the first pattern of the captured sequence and a quadratic code is used to estimate the mixture between two consecutive unstructured light patterns. For a given set of n

patterns, a linear codeword is n bits for n bits of information and a quadratic codeword is $\frac{n^2-n}{2}$ bits providing $n \log n$ bits of information, as explained in [34]. To illustrate, consider an example of 60 patterns, a linear codeword is 60 bits for 60 bits of information and a quadratic codeword is 1770 bits for 354 bits of information. Thus, the quadratic code increases the amount of information and reduces the LSH matching errors. By increasing the number of bits, the quadratic code increases the number of transitions (0 to 1 or 1 to 0) between neighboring pixels by a factor $\log n$ (in our example, $\frac{354}{60} \approx 6$). This increases the subpixel accuracy since it relies on those bit transitions.

5.4 Subpixel accuracy

In order to establish the pixel correspondences between the camera and the projector, an unsynchronized unstructured light method is used [16]. Because this method provides bidirectionality of the matches (camera to projector and projector to camera), our method will achieve subpixel accuracy in both directions. For simplicity, only the process of estimating the subpixel correspondences from the projector to the camera will be described. As explained in the previous section, subpixel matching assumes that a projector pixel is observing a mixture of two adjacent pixels in the camera image. This mixture can be described by the parameters (δ_x, δ_y) which represent a non integral displacement from an original integer match $(\hat{x}, \hat{y})$.

5.4.1 Selecting the right quadrant

Before finding the subpixel camera position for any projector pixel, the discrete projector to camera correspondence must be established by using the LSH algorithm. We thus start with a discrete match between projector pixel $\mathbf{p}'$ and camera pixel $\mathbf{p} = (\hat{x}, \hat{y})$ to which a subpixel displacement (δ_x, δ_y) is added to yield the exact match. To estimate the subpixel displacement (δ_x, δ_y), it is necessary to select the quadrant which contains pixel $\mathbf{p}$ and its three neighboring pixels. The subpixel position $(\hat{x} + \delta_x, \hat{y} + \delta_y)$ is located between those

four pixels of the camera which are represented by

$$x \leq \hat{x} + \delta_x = x + \lambda_x < x + 1, \quad x = \lfloor \hat{x} + \delta_x \rfloor \tag{5.1}$$

$$y \leq \hat{y} + \delta_y = y + \lambda_y < y + 1, \quad y = \lfloor \hat{y} + \delta_y \rfloor \tag{5.2}$$

so we can represent the subpixel position $(\hat{x}+\delta_x, \hat{y}+\delta_y)$ as $(x+\lambda_x, y+\lambda_y)$ where $0 \leq \lambda_x < 1$ and $0 \leq \lambda_y < 1$.

Because the chosen approach uses the unsynchronized unstructured method, it is possible that the projected patterns are mixed temporally in the camera image. This mixture is always computed individually for each camera pixel. For the case of subpixel matching from projector to camera, the four camera pixels forming the quadrant will each feature a different temporal mix. In the case of camera to projector matching, a single mixture value will be shared by the four projector pixels forming the quadrant. In all cases, the temporal mixture must be applied before a spatial interpolation in order to obtain accurate subpixel matches.

5.4.2 Estimating the subpixel position

The subpixel position (λ_x, λ_y) is located inside the region between the four selected neighboring pixels $\{(x,y),(x+1,y),(x,y+1),(x+1,y+1)\}$. Image intensities will be derived through bilinear interpolation over the quadrant with the parameters (λ_x, λ_y), defined as :

$$I[x+\lambda_x, y+\lambda_y] = (1-\lambda_y)I[x+\lambda_x, y] + \lambda_y I[x+\lambda_x, y+1] \tag{5.3}$$

where

$$I[x+\lambda_x, y] = (1-\lambda_x)I[x,y] + \lambda_x I[x+1,y] \tag{5.4}$$

with $0 \leq \lambda_x, \lambda_y < 1$.

In order to obtain a binary code, we select a number of image pairs from the temporal sequence and subtract them to get intensity differences. These intensities are then binarized to provide the binary code used by LSH for matching.

$$V[x,y] = I_i[x,y] - I_j[x,y] \qquad \forall (i,j) \text{ selected image pairs} \tag{5.5}$$

The intensity difference vector V is then binarized into the code C as

$$C[x,y] = binarize(V[x,y]) \tag{5.6}$$

where $binarize(x)$ is 1 if $x > 0$, 0 if $x < 0$ and a random sample from $\{0,1\}$ when $x = 0$.

The idea for subpixel matching is that the camera code will best match a projector code which is obtained from image intensities which are interpolated according to the subpixel position. In practice, codes are quantized so they change in steps, which is hard to minimize. By using the non quantized vectors $V[x+\lambda_x, y+\lambda_y]$, the cost can be made continuous and easier to minimize using gradient descent.

5.4.3 From binary cost function to continuous cost function

Instead of quantizing the pattern intensity differences V into a binary code C, we directly use V to compute the subpixel value. Two vectors are calculated; the first one, V represents the intensity differences of the pixel $\mathbf{p}$ while the second one, V', which is a reference vector, representing the corresponding coding intensities of the pixel $\mathbf{p}'$.

The subpixel optimization will minimize the angle between vectors V and V', so the objective function is simply defined as

$$cost[x+\lambda_x, y+\lambda_y] = angle(V[x+\lambda_x, y+\lambda_y], V'[x,y]) \tag{5.7}$$

where

$$angle(a,b) = \arccos\left(\frac{a \cdot b}{\|a\|\|b\|}\right) \tag{5.8}$$

In practice, for simplicity, we do not compute the inverse cos and change this angle function to approximately return the number of bit transitions:

$$angle(a,b) = \left(1 - \left(\frac{a \cdot b}{\|a\|\|b\|}\right)\right) * \frac{n}{2} \tag{5.9}$$

where n is the number of bits in the code. This cost has a minimum of 0 when a and b are aligned (corresponding to an angle 0°), an average of $n/2$ bits when the angle is 90°, when

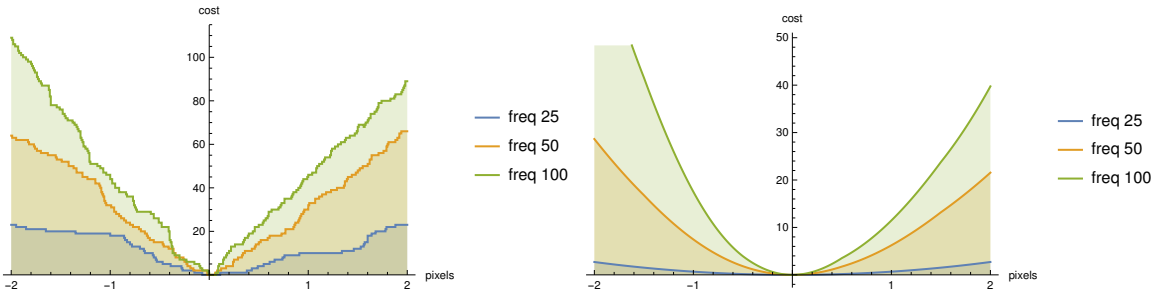


Figure 5.4: Pattern frequency representing the number of oscillations of one sine in an unstructured light pattern. The blue, orange and green curves correspond to a frequency of 25, 50 and 100 oscillations, respectively. These curves are a cost function of the difference between neighboring pixels. The curves (left) represent a binary difference between the pixel codes and the curves (right) represent a continuous difference of two vectors consisting of pixel intensities.

vectors are uncorrelated, and a maximum of n when the vectors are inversely correlated at $180°$.

The optimization estimates the subpixel match by minimizing the cost over possible δ_x and δ_y, starting at discrete position $(\hat{x}, \hat{y})$.

Fig. 5.4 illustrates the difference between a binary cost function and a continuous cost function. Binary cost function curves feature steps where the gradient is 0. In the continuous cost function, the curves are much smoother and precise, so they are better to be optimized on and the gradient descent can easily find the minimum.

5.4.4 Gradient Descent

As explained above, we used a gradient descent to reduce the computation time for the subpixel search and increase its accuracy. Gradient descent iteratively converges to the local minimum of a function following the negative direction of the gradient at a current point. We minimize the cost for the angle between the two vectors, explained above in Sec. 5.4.3. The obtained curve is a bowl-shaped curve. Our cost function lends itself well

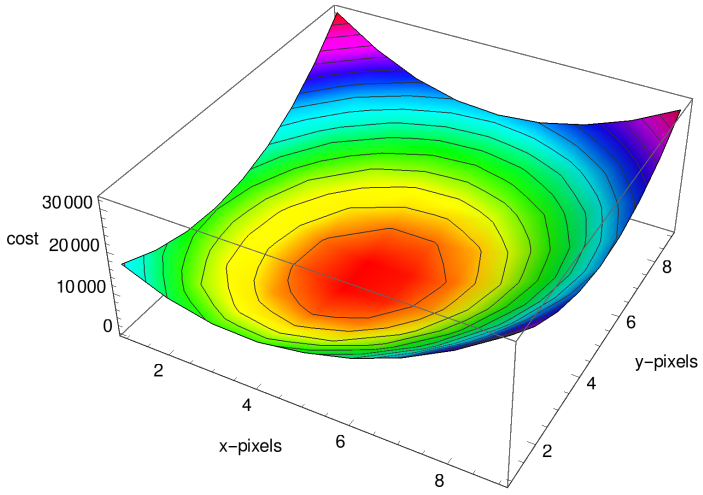


Figure 5.5: The x, y, and z axis represent the x, y pixels and the cost of the difference between neighboring pixels, respectively. We try to minimize this cost function curve.

to the minimization due to its shape, as shown in Fig. 5.5, as it is locally convex, as required by the gradient descent algorithm.

5.4.5 Correcting match errors

An important property of unstructured light patterns is the correlation of the neighboring pixels. On the contrary, there is no correlation between two distant pixels because the patterns are generated randomly. Fig. 5.4 illustrates the two parts of our cost function and displays at which point is there no more correlation between pixels. Using LSH to establish the pixel correspondences between the camera and the projector generates several matching errors featuring a small deviation from the correct match. The subpixel computation can correct these matching errors, if the corresponding pixel is part of the neighborhood where pixels are correlated. However, if there is no correlation then the subpixel cannot find the correct match. Thus, LSH errors can be compensated by our subpixel method in some cases, namely local matching errors.

For the sake of illustration, the same reference patterns were matched twice adding a

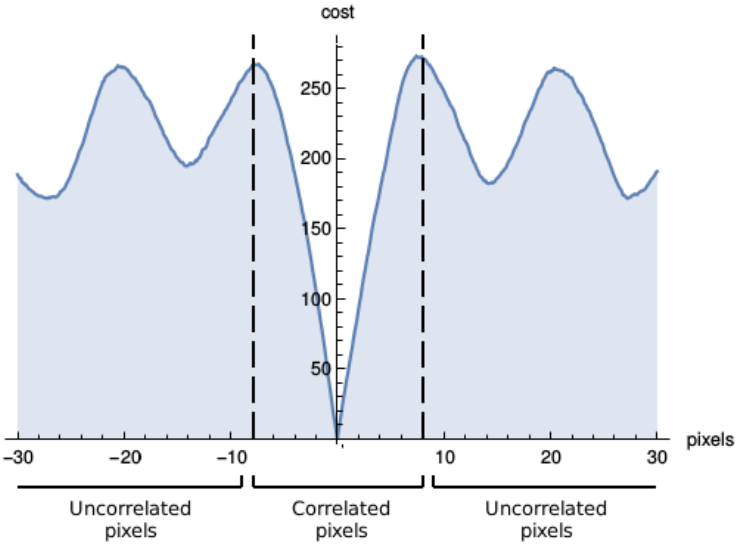


Figure 5.6: Cost function curve which shows that within a specific neighborhood, ± 10 pixels in this case, in the unstructured light pattern, the cost is monotonous and easy to minimize.

noise (± 4 randomly to each matched pixel), a first time without subpixel and a second time with the subpixel matching. This noise generates a lot of LSH errors. Fig. 5.7 illustrates the improvement of the matches.

In addition, if the frequency is very low then the subpixel can improve and correct the matches because the correlated neighborhood is wider. On the other hand, if the frequency is very high, the subpixel has a small area of convergence and can no longer correct large matching errors (see Fig. 5.2). An example where this matters is if you want to scan faces. In this case, there is an upper limit to the usable frequency since skin presents subsurface scattering which blurs high frequencies. Nevertheless, our subpixel method can compensate for the matching errors and increase the accuracy and the quality of matches.

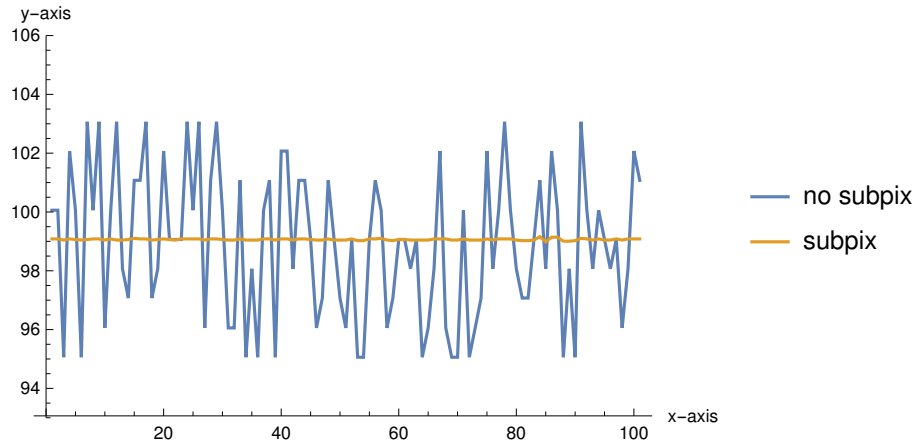


Figure 5.7: Comparison of matches between matching twice the same reference patterns adding a random noise. The blue curve represents a correspondence without a subpixel accuracy and the orange curve represents a subpixel correspondence. Subpixel accuracy can improve and correct the matching errors in the area where pixels are correlated.

5.5 Experiments

This section presents various experiments to evaluate our method in real scenes as well as compare it to other methods. Furthermore, the experimental setup used to achieve these experiments is described. Finally, two sets of results are provided; quantitative results to compare subpixel accuracy between our method and other methods, and qualitative results to compare the quality of 3D models generated by different methods.

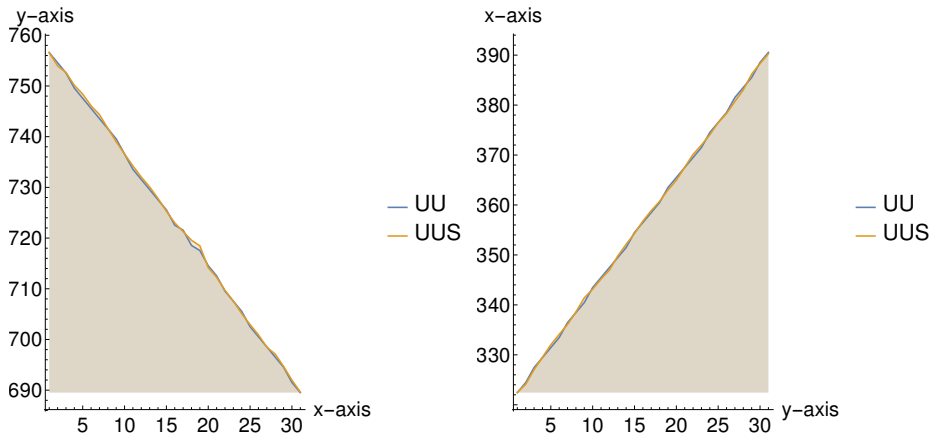
In all the experiments, common off-the-shelf equipment is used. The camera is a raspberry PI at a resolution of 1280x720 and the projector is an Aaxa HD Pico projector at its native resolution of 1280x720. The projection and the capture are accomplished at 30 fps. Many difficulties were encountered with this common material such as the *auto gain*, the *auto focus* and *flicker*. *Auto gain* is the automatic brightness adjustment of the camera to the illumination of the scene. *Auto focus* is the focus done automatically by the camera to the scene depths. This can thus change the calibration. Finally, flicker is the mixture of

colors that the camera sees. To project an RGB image, most RGB projectors send one color at a time, and should the camera have a very short exposure time, then it can distinguishe a mixture of each color. Thus, it is no longer possible to triangulate and obtain 3D models. The camera-projector system was calibrated with a simple planar calibration [57, 43]. In addition, our experiments were performed in difficult conditions with a rolling shutter camera.

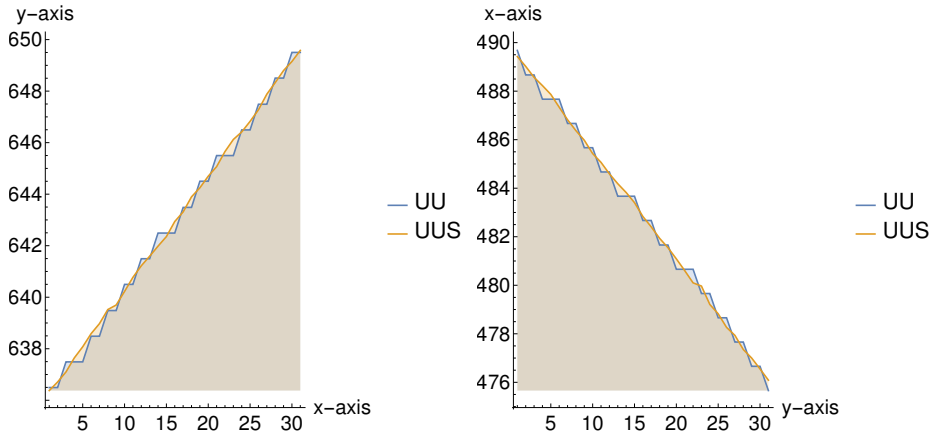
To evaluate the proposed method, it is compared to the unsynchronized unstructured method without subpixel [16] and to the *Phase Shift* method [47]. In our experiments, a looping video of 60 unstructured light patterns is projected at 30 fps without synchronization between the projector and the camera. Furthermore, in order to unwrap the phase for the *Phase Shift* method, 16 patterns of a shifted sine (8 patterns for each axis) are added to the 60 unstructured light patterns. The decoding step is performed with the unstructured light patterns then the subpixel is computed from the recovered phases. Because the video is projected and captured at 30 fps, it is important to find the mixture between two consecutive patterns using the unsynchronized unstructured light method.

In this section are presented a first set of results which consist of a quantitative comparison between the three methods, then a second set which consists of a qualitative comparison. The experiments are accomplished on different real scenes; a plane, a specular corner and a Lambertian robot. The results presented above are the raw data obtained, no median filter or equivalents were applied. For the calculation of the phase in each period, a treatment is performed on the neighboring points to unwrap the phase. Then, for the triangulation of the 3D models, a selection of the 3D points is carried out to remove the outliers or the points with an aberrant depth ($z = \pm 200$), and this for the three methods.

The first experiment is to compare unsynchronized unstructured light methods with and without subpixel accuracy. For this experiment, 60 unstructured light patterns are projected on a plane with a pattern frequency of 50 (number of cycles per image). The pixel ratio of this experiment is equal to 2 (each camera pixel sees 4 neighboring projector pixels, thus 2 pixels per axis). Fig. 5.8 presents a comparison of the two methods. In this figure,



(a)



(b)

Figure 5.8: The curves represent a line extracted from two LUTs; (a) the camera view and (b) the projector view (b). The blue curve represents the unsynchronized unstructured light method without the subpixel accuracy (UU) and the orange line represents the unsynchronized unstructured light method with the subpixel accuracy (UUS). The figures left and right represent a number of pixels along the x and y axis, respectively.

from the projector view, the addition of subpixel precision improves the curve by making it smoother as compared to its counterpart, without subpixel, which has a step function shape. On the other hand, from the camera view, the improvement is minimal because of the pixel ratio. One can say that the camera-projector correspondence already has some level of subpixel accuracy.

For the second experiment, 60 unstructured light patterns and 16 patterns of a shifted sine are projected on a specular corner using a frequency of 50. Furthermore, the same pixel ratio (near 2) has been kept. Fig. 5.9 (left) shows the curves of the three methods from the camera view; unsynchronized unstructured light method without and with the subpixel accuracy and the *Phase Shift* method. Fig. 5.9 (right) illustrates the average error of each method. The average error is the difference between a line extracted from the LUTs and the reference line. One can notice that there is a slight improvement in the unsynchronized unstructured light method curve with subpixel compared to that without subpixel accuracy. One can further notice that the error curve of the *Phase Shift* method is shifted about 4 pixels because of the specular surface of the reconstructed object.

For the third experiment, the scans are accomplished at different frequencies. As explained in Sec. 5.3.1, the pattern frequency has a significant impact on subpixel accuracy. Fig. 5.10 shows a comparison between the unsynchronized unstructured light method with and without subpixel accuracy. The pattern frequency of each scan is (a) 25, (b) 50, (c) 70. It can be seen that the blue curves with the frequencies 25 and 50 are of step function shape. The curves of the subpixel unsynchronized unstructured light method are much smoother and have no steps. The subpixel corrects even some matching errors because the cost function curve is wider (Fig. 5.4, freq 25 and 50), so the neighboring pixels are correlated over a larger zone (Fig. 5.6). On the other hand, the curve with a frequency 70 is less smooth because the cost function curve is very pronounced and the correlation zone is very small. The mean and the standard deviation show that the scan at a frequency 70 is better but that the subpixel cannot improve it more as is the case of the frequencies 25 and 50, as shown in Table 5.1.

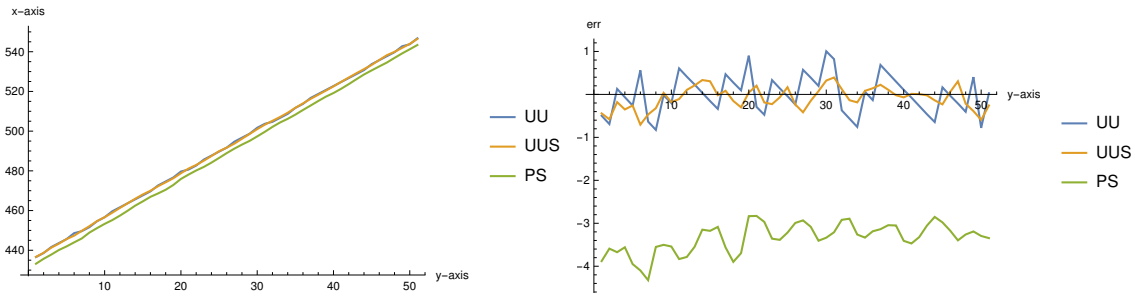


Figure 5.9: The curves (left) represent a line extracted from three LUTs; the blue curve represents the unsynchronized unstructured light method without the subpixel accuracy (UU), the orange line represents the unsynchronized unstructured light method with the subpixel accuracy (UUS) and the green curve represents the *Phase Shift* method (PS). The curves (right) represent the average error between the extracted line and a reference line passing through all the points.

The last experiment in the quantitative results set is the comparison of different pixel ratios. In this experiment, the camera view is chosen and the pattern frequency used is 50. The pixel ratio represents the number of pixels matched between the camera and the projector. We chose three different pixel ratios to demonstrate the achievements of the subpixel accuracy; a camera pixel sees only one projector pixel (ratio = 1), a camera pixel sees 4 projector pixels so 2 pixels per axis (ratio = 2) and finally a camera pixel sees 16 projector pixels so 4 pixels per axis (ratio = 4). Table 5.2 illustrates the results of the unsynchronized unstructured light method and the subpixel unsynchronized unstructured light method. Mean and standard deviation represent the difference between a line extracted from a LUT and a reference line. The quality of the matches improves when the pixel ratio increases (the average error and the standard deviation decrease). On the other hand, the higher the ratio, the less the subpixel improves the quality as one can say that the correspondence is already subpixel.

For the set of qualitative experiments, four 3D reconstructions obtained with the subpixel unsynchronized unstructured light method and the *Phase Shift* method are presented.

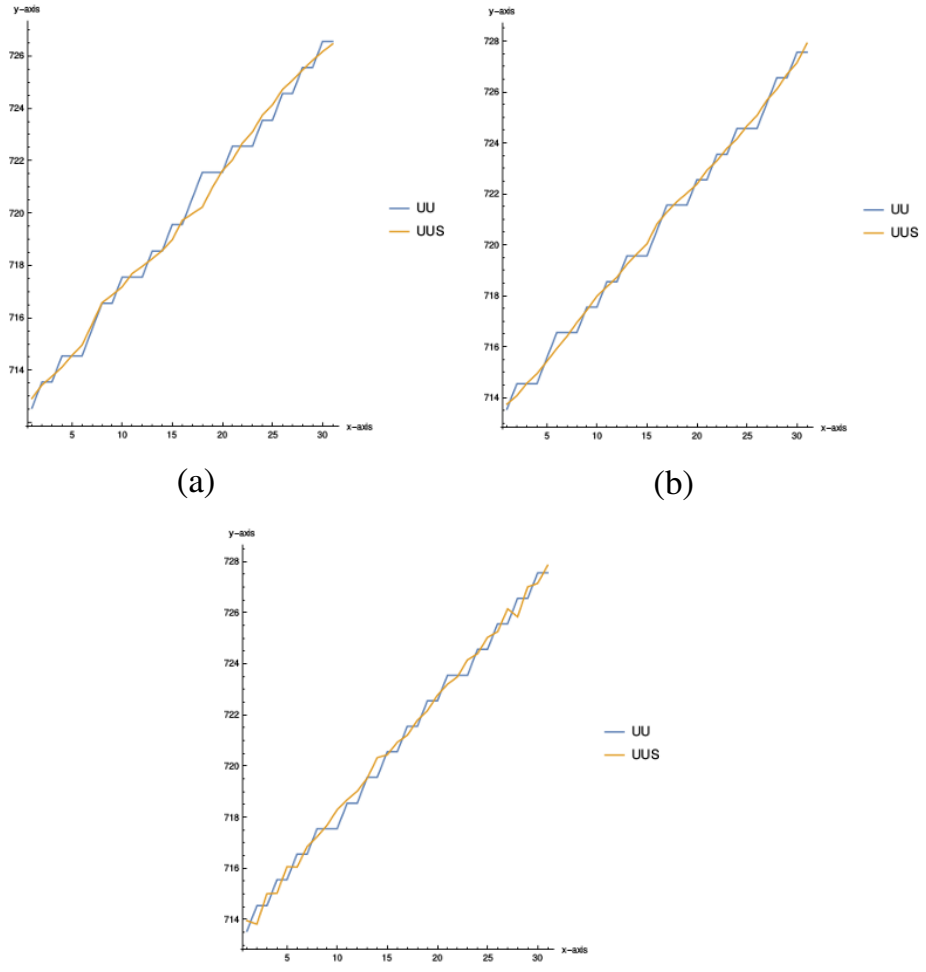
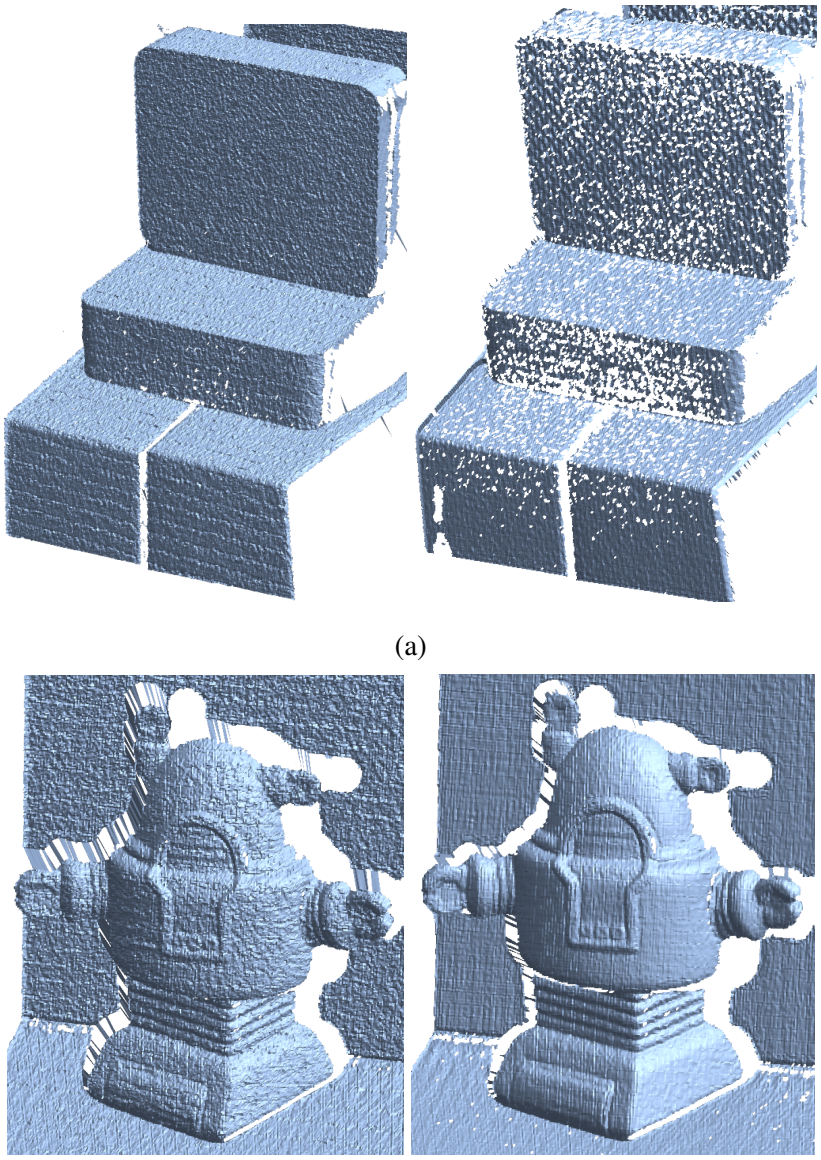


Figure 5.10: Extracted line from two LUTs of an unstructured light pattern projection with a frequency of (a) 25, (b) 50, (c) 70; where the frequency represents the number of cycles of each sine per pattern. The blue curve represents the unsynchronized unstructured light method without the subpixel accuracy (UU), the orange line represents the unsynchronized unstructured light method with the subpixel accuracy (UUS).



(a)

(b)

Figure 5.11: Various scenes reconstructed in 3D. (a) shows a 3D reconstruction of a specular corner (a right angle) and (b) shows a 3D reconstruction of a Lambertian robot. The 3D reconstructions (left) are obtained using the unsynchronized unstructured light method with the subpixel precision and the 3D reconstructions (right) are obtained using the *Phase Shift* method. These unfiltered models are obtained from the camera view.

freq	subpixel	Mean	std
25	without	0.255	0.167
	with	0.163	0.112
50	without	0.241	0.169
	with	0.082	0.065
70	without	0.225	0.123
	with	0.140	0.128

Table 5.1: The standard deviation of the difference (in pixels) between a reference line and an extracted line from each LUT in x obtained with a different pattern frequency for each set of unstructured light patterns. Mean and std represent the mean and standard deviation for unsynchronized unstructured light methods with and without subpixel accuracy, respectively.

Fig. 5.11 (a) shows a specular corner and Fig. 5.11 (b) shows a Lambertian robot. The *Phase Shift* model (right (a)) has several holes due to matching errors. These matching errors generate outliers that are removed during the step of calculating polygons to form a 3D model. As a result of the previously mentioned errors, the quality of the matches of the subpixel unsynchronized unstructured light method is deemed superior to the quality of the matches of the *Phase Shift* method. This is because the corner is specular and there is also a mixture between two unstructured light patterns due to the unsynchronized capture. The subpixel unsynchronized unstructured light method is robust to specular objects and to the unsynchronized capture, as shown in Fig. 5.11 (a) and (b) on the left.

Fig. 5.12 shows a 3D model achieved with the proposed method from the projector view. The cropped image (left) shows more details, obtained through the subpixel precision, than the cropped image (right) which is achieved without subpixel. Fig. 5.13 illus-

ratio	subpixel	Mean	std
1	without	0.190	0.133
	with	0.088	0.122
2	without	0.148	0.112
	with	0.109	0.084
4	without	0.081	0.059
	with	0.057	0.053

Table 5.2: The standard deviation of the difference (in pixels) between a reference line and an extracted line from each LUT in x obtained with a different pixel ratio for each set of unstructured light patterns. Mean and std represent the mean and standard deviation for unsynchronized unstructured light methods with and without subpixel accuracy, respectively.

trates a section of the 3D model (robot). It shows the accuracy of each method on a section of the robot. The quality of the reconstruction is very good and more details can be noticed with subpixel unsynchronized unstructured light and the *Phase Shift* methods.

The goal of this method is to quickly and efficiently scan faces. In addition to scanning in less than two seconds, the accuracy of the matches is increased by adding subpixel. Fig. 5.14 illustrates a 3D model of a face from the projector view. An excellent 3D model with the utmost precision is obtained using the proposed method.

5.6 Conclusion

In this article, we proposed a new method to achieve high subpixel accuracy using the unsynchronized unstructured light method. This method increases the precision of the correspondence between the projector and the camera. The unsynchronized unstructured light

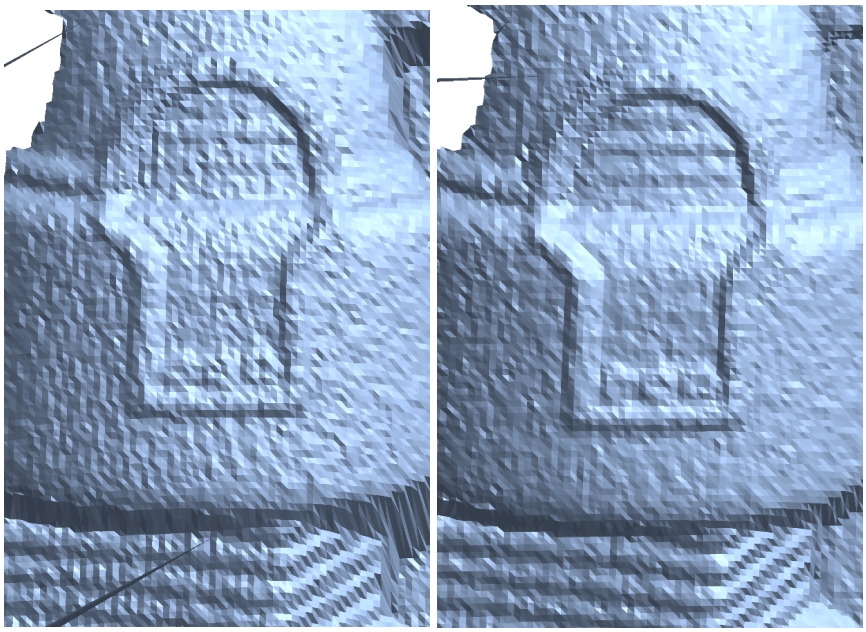


Figure 5.12: Reconstruction of a Lambertian robot. 3D models are obtained with the unsynchronized unstructured light method without subpixel accuracy (left) and with subpixel accuracy (right). This unfiltered model is obtained for the projector view.

method makes scanning faces easier in difficult conditions such as subsurface scattering, indirect illumination and scene discontinuities. Relying on low cost hardware without any form of temporal synchronization and a high frame rate, at 30 fps and 60 fps, 3D models with the utmost precision can be achieved. The subpixel estimation is fast and simple, and can also correct errors of the discrete correspondences for a better match quality.

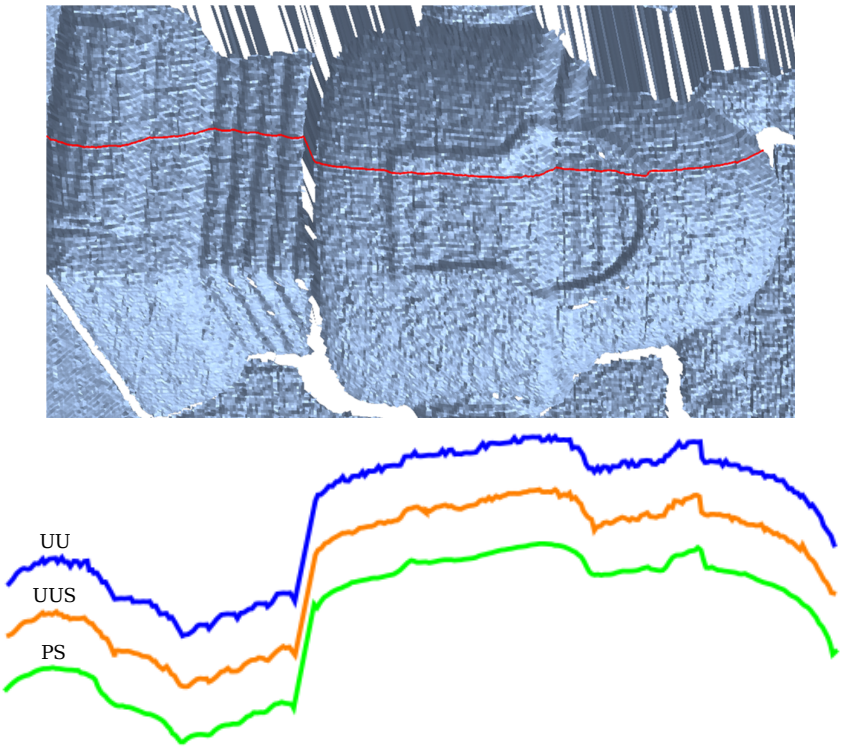


Figure 5.13: x and y projection (bottom) of reconstructed Lambertian robot for different methods. The blue curve represents the unsynchronized unstructured light method without the subpixel accuracy (UU), the orange line represents the unsynchronized unstructured light method with the subpixel accuracy (UUS) and the green line represents the *Phase Shift* method. The figure (top) illustrates the portion of the robot which is reconstructed.

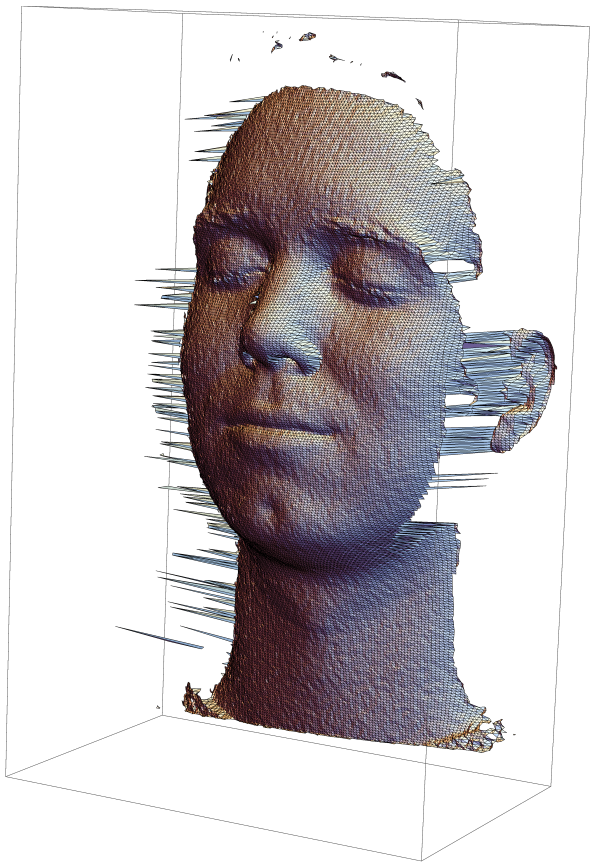


Figure 5.14: 3D reconstruction using subpixel unsynchronized unstructured light method of a face. This unfiltered model is obtained from the projector view.

Chapitre 6

UNE APPLICATION: CASQUE DE RÉALITÉ VIRTUELLE

Les dernières années ont vu l'apparition des technologies de réalités virtuelles chez les particuliers notamment les casques de VR (*Virtual Reality*) à des prix plus abordables. Ce chapitre décrit la conception et la fabrication d'un casque VR personnalisé. Suite à la réalisation d'un scanner 3D à lumière non structurée non synchronisé pour pouvoir scanner les visages, il est nécessaire de s'interroger sur les applications potentielles de cette technologie. L'exemple exploré dans ce présent mémoire est la fabrication d'un casque VR personnalisé épousant les contours du visage de l'utilisateur. Dans l'optique d'assurer la faisabilité du projet, cette partie a été effectuée avant la réalisation du scanner 3D décrit précédemment. De ce fait, le scanner 3D utilisé est une *Kinect 2* afin d'obtenir rapidement un prototype du casque VR.

L'étape initiale consiste au calibrage de la *Kinect 2* afin de maximiser la précision du modèle 3D du visage. La *Kinect 2* génère en sortie une carte de profondeur pour la reconstruction 3D. C'est une image qui encode tous les niveaux de profondeur de la scène à scanner. Chaque niveau de profondeur est représenté par une couleur comme l'illustre la Fig. 6.1. Les hautes et les basses intensités sont représentées par la couleur bleue et la couleur verte, respectivement. A priori, il faut calibrer la Kinect pour pouvoir récupérer les bons niveaux de profondeur et dans ce cas-ci, la méthode choisie est le calibrage avec un objet 3D connu. Un minimum de six points visibles dans le monde est nécessaire pour effectuer la calibration. L'objet de calibration dans le cas présent est un carré sur un plan qui est déplacé à l'aide d'un rail millimétrique. De cette façon, un cube virtuel à huit sommets visibles est généré avec le rail. Il suffit donc de mettre en correspondance ces points 3D visibles et les coordonnées en pixels récupérées dans l'image capturée. L'étape suivante est de retrouver l'homographie H qui représente la transformation géométrique entre les

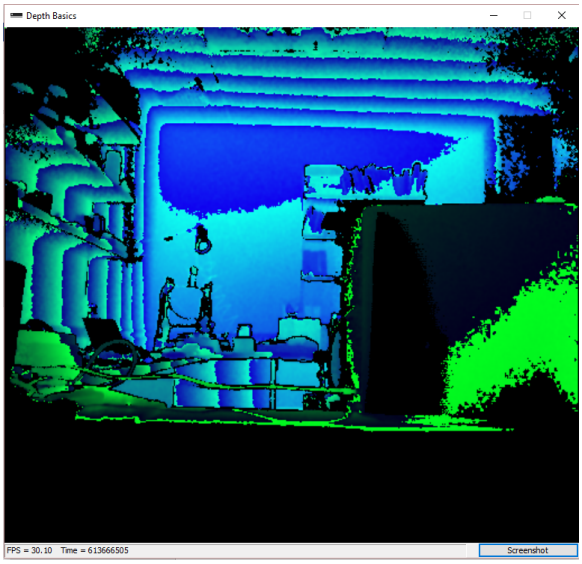


Figure 6.1: Une carte de profondeur générée par la *Kinect 2*. Les couleurs bleue et verte représentent les hautes et les basses intensités (la variation de la profondeur), respectivement.

points dans le monde et dans l'image. Avec cette homographie, nous pouvons récupérer les valeurs des points 3D avec leurs vraies profondeurs.

Suivant le calibrage vient la réalisation d'un scan de visage et sa reconstruction 3D. À partir de la carte de profondeur obtenue de la Kinect, un nuage de points 3D est généré. Dû au type de calibrage choisi, le calibrage avec un objet 3D, le modèle 3D obtenu n'est pas centré à l'origine du monde. L'idée est de remettre droit le modèle en prenant un triangle de référence et de l'aligner selon les trois axes. La Fig. 6.2 (à gauche) représente le résultat obtenu d'un nuage de points 3D suite à son alignement. Comme expliqué dans le chapitre 4, les points ont été reliés en forme de triangles afin de récupérer un beau modèle 3D, tel qu'illustré dans la Fig. 6.2 (à droite).

Enfin vient l'étape de la réalisation du casque 3D, étape dite de modélisation. Ici, les *courbes B-Spline* sont choisies afin de sélectionner un contour sur le modèle en 3D.Les *courbes B-Spline* sont une généralisation des *courbes de Bézier*. Ces dernières sont basées

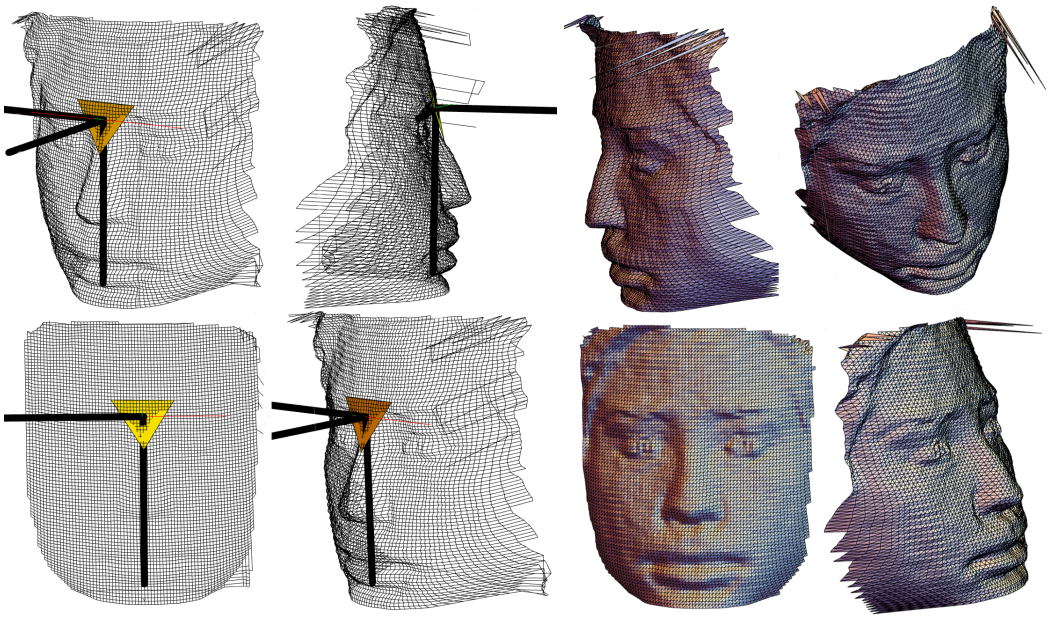


Figure 6.2: Nuage de points (gauche) remis à l'origine du monde à l'aide d'un triangle de référence sous quatre vues différentes. Ce triangle de référence est calculé à partir de trois points sélectionnés dans le visage. Le modèle 3D (droite) résultant de la triangulation du nuage de points sous quatre vues différentes.

sur les *polynômes de Berstein*, tel que:

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!}u^i(1-u)^{n-i} \tag{6.1}$$

Ce sont des courbes paramétriques :

$$C(u) = \sum_{i=0}^{n} B_{i,n}(u)P_i, 0 \leq u \leq 1 \tag{6.2}$$

Les *courbes de Bézier* présentent plusieurs propriétés. Elles sont à l'intérieur des enveloppes convexes. Ces enveloppes sont formées par des points de contrôle. De plus, ces courbes sont dérivables jusqu'à $n-1$ des points de contrôle. Leur avantage est l'invariabilité aux transformations affines telles que la translation, la rotation, etc. Toutefois, si un seul

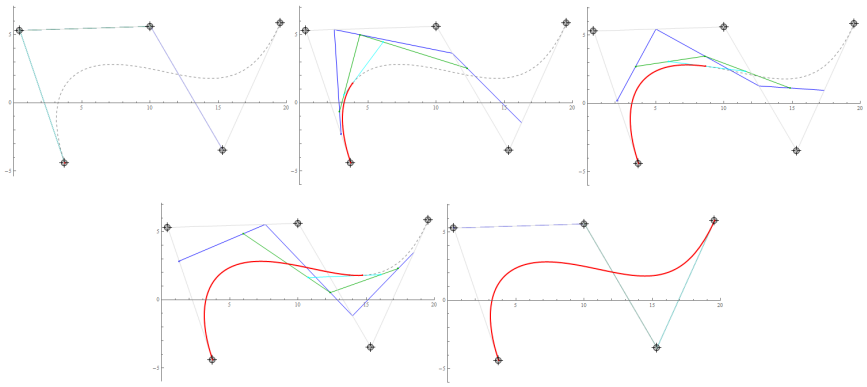


Figure 6.3: Différentes étapes de la construction de la Courbe de *Bézier*. La courbe rouge représente la Courbe de *Bézier* à l'intérieur de son enveloppe formée par des points de contrôle. Les courbes bleues, vertes et turquoises représentent les différentes tangentes à la Courbe de *Bézier*.

point de contrôle est modifié alors toute la courbe change. Les courbes peuvent être contrôlées globalement comme indiqué dans la Fig. 6.3. Les *courbes B-Spline* non uniformes bénéficient des mêmes propriétés que les *courbes de Bézier*. Par contre, leur avantage est qu'il est possible de les contrôler localement. Ainsi, il est possible d'interpoler entre les points de contrôle. C'est pour cela que les *courbes B-Spline* ont été choisi afin de pouvoir récupérer la forme de la surface du modèle 3D.

L'objectif est de tracer une courbe sur la surface et de récupérer la position des points de contrôle correspondants. Afin de minimiser le taux d'erreur et de simplifier le problème, une approximation sur la symétrie horizontale du visage pour former la *courbe B-Spline* est posée. En effet, il est possible de sélectionner seulement sur une partie du visage et appliquer une réflexion par rapport à l'axe y sur les points de contrôles obtenus (Fig. 6.2 gauche). Ensuite, il ne suffit que de calculer les points de la *courbe B-Spline* à l'aide de ces

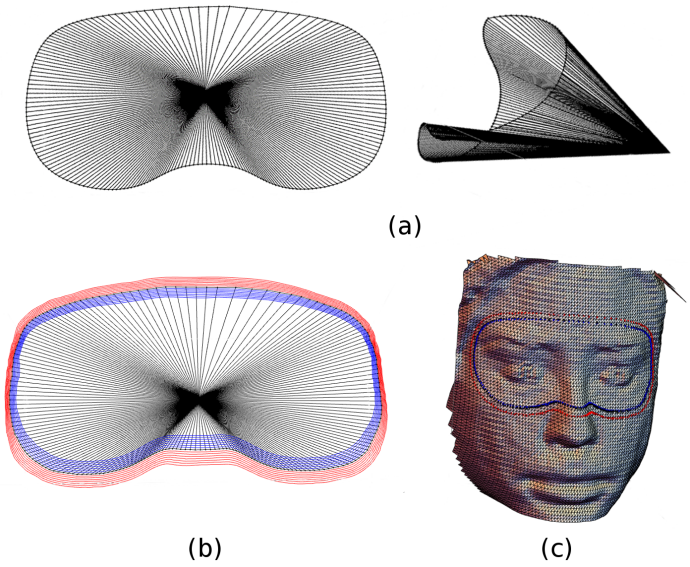


Figure 6.4: Les points de la *courbe B-Spline* (a) sont reliés par un point de référence (origine du monde). Ce point de référence permet d'aligner les *courbes B-Spline* avec le modèle 3D du visage. Les *courbes B-Spline* rouges et bleues (b) sont générées de chaque côté de la courbe originale (noire). La vue du modèle 3D du visage (c) avec les différentes *courbes B-Spline* superposées.

points de contrôles et des polynômes afin d'obtenir une courbe dense et lisse (Fig. 6.4).

Il est nécessaire de fabriquer une surface lisse afin de poser le casque sur le visage. Pour pouvoir former cette surface à partir d'une *courbe B-Spline*, il a été nécessaire de décaler tous les points de la courbe selon un nouveau repère précalculé pour chaque point. En effet, pour chaque point de la courbe, un nouveau repère a été calculé avec le point même comme son origine et l'axe des y est perpendiculaire à la *courbe B-Spline*. Ensuite, une translation est appliquée sur les points selon l'axe y. Il est important de décaler chaque point de la courbe dans le bon sens sans perdre ou modifier les détails récupérés sur le modèle 3D. La *courbe B-Spline* est décalée plusieurs fois jusqu'à avoir une surface suffisante pour le casque 3D Fig. 6.4 (b). Ensuite, il suffit d'intersecter les courbes obtenues avec le modèle 3D du visage pour récupérer la bonne forme (voir Fig. 6.4 (c)).

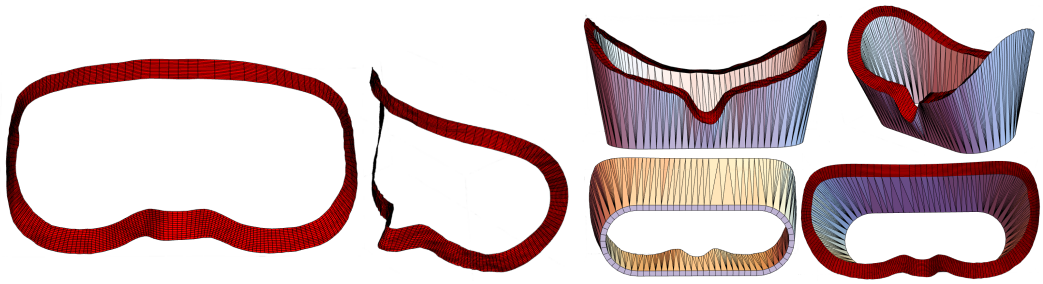


Figure 6.5: Modèle 3D résultant de la triangulation des *courbes B-Spline* sous deux vues différentes (gauche). Et le modèle 3D final du casque VR sous quatre vues différentes (droite).

Figure 6.6: Une version imprimée en plastique *NinjaFlex* d'un casque VR pour le visage de Chaima El Asmi.

La même méthode pour fermer le modèle est utilisée que précédemment. Une surface lisse et dense est obtenue et celle-ci épouse parfaitement la forme du visage 3D (voir Fig. 6.5 (gauche)). Une fois la surface obtenue, il faut maintenant terminer le modèle du casque. Pour cela, il est nécessaire de connaître la position des yeux de l'utilisateur pour former l'autre extrémité du casque à une distance idéale pour le focus de l'individu. Ainsi, chaque personne a une hauteur du casque ajustée à sa hauteur des yeux. Le résultat final du

casque 3D est la combinaison de la surface lisse (Fig. 6.5 (gauche)) et d'une surface ovale formée au niveau des yeux. La Fig. 6.5 (droite) représente la version finale du casque 3D.

Une fois la reconstruction 3D réalisée, il a été possible de construire le casque VR personnalisé à la géométrie du visage et d'imprimer le résultat à l'aide d'une imprimante 3D de type *MendelMax 2*. Le modèle imprimé a été testé afin de valider qu'il correspond fidèlement à la forme du visage de la personne scannée, illustré dans la Fig. 6.6. Toutefois, la *kinect 2* employée ne permettait pas d'obtenir une résolution du visage désirée. Une perte de détail importante a été remarquée. Ce résultat démontre ainsi le besoin d'un scanner 3D rapide et précis pour l'application de scans de visages.

CONCLUSION

Ce mémoire s'intéresse à deux problèmes majeurs dans le domaine des reconstructions actives. Les travaux présentés ont pour but d'améliorer la méthode à lumière non structurée afin de pouvoir scanner des visages rapidement avec du matériel ordinaire.

En premier lieu, nous nous sommes attaqués au problème de synchronisation entre le projecteur et la caméra. En utilisant du matériel à bas coût et en projetant à une fréquence d'images élevée (30 fps et 60 fps), il est possible de réaliser des scans 3D d'objets en moins de deux secondes sans synchroniser le système projecteur-caméra. À l'aide de cette méthode, les résultats obtenus sont démontrés très proches de ceux obtenues avec la capture synchronisée. Cette méthode permet de faciliter et d'accélérer le scan des visages ainsi que les scans dans des systèmes de caméras et de projecteurs très éloignés où il serait habituellement impossible de synchroniser. De plus, cette méthode se veut accessible aux utilisateurs qui n'ont pas accès à du matériel industriel ou de fine pointe. Elle leurs permet d'obtenir facilement des modèles 3D denses de qualité.

En deuxième lieu, nous avons visé à améliorer la qualité des correspondances en ajoutant du sous-pixel à l'algorithme de correspondance. Le sous-pixel est un facteur important dans la reconstruction 3D, car il augmente la précision et le niveau de détails. De plus, le sous-pixel permet de corriger les erreurs de correspondances afin d'obtenir une qualité de correspondance supérieure. Dans la méthode proposée, l'estimation du sous-pixel est facile et rapide. Nous avons obtenu des modèles 3D à très haute précision dans des conditions difficiles tels que l'illumination indirecte, les scènes discontinuités ou l'utilisation d'un matériel non professionnel. Il a été possible de comparer les modèles réalisés à d'autres modèles réalisé par des méthodes classiques. Aussi, une évaluation est effectué sur des différents paramètres qui affectent le sous-pixel tels que le ratio de pixels et la fréquence spatiale des patrons.

En troisième lieu, la méthode proposée offre une robustesse et une précision équivalentes sinon meilleures que les méthodes classiques, mais avec une plus grande rapidité et robustesse. Les scanners 3D sont omniprésents dans divers domaines depuis plusieurs années. Ainsi, nous pensons que ce scanner 3D se compare avantageusement avec ceux disponibles sur le marché et ce à cause de leur prix onéreux ou de leur longue durée de scan. Il existe diverses applications aux scanners 3D, l'application à l'étude dans ce mémoire est le casque personnalisé de réalité virtuelle, qui épouse parfaitement les contours du visage grâce au scan 3D rapide du visage. Avec le modèle précis du visage qui est obtenu, il est d'ailleurs possible de fabriquer et imprimer un casque VR sur mesure qui procure un grand confort.

Plusieurs perspectives de recherches futures restent à explorer dans le domaine des scanners 3D. L'une de ces perspectives touchant notre approche à lumière non structurée non synchronisée sous-pixel est le scanner à main libre. Notre matériel actuel est attaché sur un support amovible et doit rester immobile pendant au moins deux secondes afin de réussir à scanner des scènes statiques. Dans le futur, il est envisageable de mettre au point un scanner 3D qui peut scanner des scènes dynamiques ou scanner "main libre".

RÉFÉRENCES

[1] Calibration OpenCV. https://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html#cameracalibrationopencv.

[2] Google Project Tango. https://get.google.com/tango/.

[3] Optical Society of America. https://www.osapublishing.org/aop/abstract.cfm?URI=aop-3-2-128.

[4] Sensor Structure. https://structure.io/.

[5] Triangulation OpenCV. https://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html?highlight=triangulation.

[6] Wikipédia, Scanner 3D. https://fr.wikipedia.org/wiki/Scanner_tridimensionnel.

[7] Alexandr Andoni et Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Dans *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 459–468. IEEE, 2006.

[8] Joan Batlle, E Mouaddib, et Joaquim Salvi. Recent progress in coded structured light as a technique to solve the correspondence problem: a survey. *Pattern recognition*, 31(7):963–982, 1998.

[9] Dirk Bergmann. New approach for automatic surface reconstruction with coded light. Dans *Remote sensing and Reconstruction for three-dimensional objects and scenes*, volume 2572, pages 2–10. International Society for Optics and Photonics, 1995.

[10] François Blais, Jean-Angelo Beraldin, Sabry El-Hakim, et Guy Godin. New development in 3d laser scanners: From static to dynamic multi-modal systems. Dans *6th Conf Opt 3-D Meas Tech*, pages 22–26, 2003.

[11] Tongbo Chen, Hans-Peter Seidel, et Hendrik PA Lensch. Modulated phase-shifting for 3d scanning. Dans *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[12] Vincent Couture, Nicolas Martin, et Sebastien Roy. Unstructured light scanning to overcome interreflections. Dans *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1895–1902. IEEE, 2011.

[13] Vincent Couture, Nicolas Martin, et Sébastien Roy. Unstructured light scanning robust to indirect illumination and depth discontinuities. *International Journal of Computer Vision*, 108(3):204–221, 2014.

[14] James Davis, Ravi Ramamoorthi, et Szymon Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. Dans *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–359. IEEE, 2003.

[15] Chaima El Asmi et Sébastien Roy. Subpixel unsynchronized unstructured light. Manuscript submitted for publication.

[16] Chaima El Asmi et Sébastien Roy. Fast unsynchronized unstructured light. Dans *Computer and Robot Vision (CRV), 2018 15th Conference on*. IEEE, 2018.

[17] Munther A Gdeisat, David R Burton, et Michael J Lalor. Eliminating the zero spectrum in fourier transform profilometry using a two-dimensional continuous wavelet transform. *Optics Communications*, 266(2):482–489, 2006.

[18] Jinwei Gu, Toshihiro Kobayashi, Mohit Gupta, et Shree K Nayar. Multiplexed illumination for scene recovery in the presence of global illumination. Dans *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 691–698. IEEE, 2011.

[19] Jens Gühring. Dense 3d surface acquisition by structured light using off-the-shelf components. Dans *Videometrics and Optical Methods for 3D Shape Measurement*, volume 4309, pages 220–232. International Society for Optics and Photonics, 2000.

[20] Mohit Gupta et Shree K Nayar. Micro phase shifting. Dans *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 813–820. IEEE, 2012.

[21] Olaf Hall-Holt et Szymon Rusinkiewicz. Stripe boundary codes for real-time structured-light range scanning of moving objects. Dans *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 359–366. IEEE, 2001.

[22] Jungong Han, Ling Shao, Dong Xu, et Jamie Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *IEEE transactions on cybernetics*, 43(5):1318–1334, 2013.

[23] Richard Hartley et Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[24] Kyriakos Herakleous et Charalambos Poullis. 3dunderworld-sls: An open-source structured-light scanning system for rapid geometry acquisition. *arXiv preprint arXiv:1406.6595*, 2014.

[25] Jonathan M Huntley et Henrik Saldner. Temporal phase-unwrapping algorithm for automated interferogram analysis. *Applied Optics*, 32(17):3047–3052, 1993.

[26] Seiji Inokuchi. Range imaging system for 3-d object recognition. *ICPR, 1984*, pages 806–808, 1984.

[27] Idaku Ishii, Kenkichi Yamamoto, Kensuke Doi, et Tokuo Tsuji. High-speed 3d image acquisition using coded structured light projection. Dans *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 925–930. IEEE, 2007.

[28] Tobias Jaeggli, Thomas P Koninckx, et Luc Van Gool. Online 3d acquisition and model integration. Dans *PROCAMS, ICCV Workshop*, 2003.

[29] Thomas R Judge et PJ Bryanston-Cross. A review of phase unwrapping techniques in fringe analysis. *Optics and Lasers in Engineering*, 21(4):199–239, 1994.

[30] Hiroshi Kawasaki, Ryo Furukawa, Ryusuke Sagawa, et Yasushi Yagi. Dynamic scene shape reconstruction using a single structured light pattern. Dans *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. Ieee, 2008.

[31] Thomas P Koninckx et Luc Van Gool. Real-time range acquisition by adaptive structured light. *IEEE transactions on pattern analysis and machine intelligence*, 28(3):432–445, 2006.

[32] Anner Kushnir et Nahum Kiryati. Shape from unstructured light. Dans *3DTV Conference, 2007*, pages 1–4. IEEE, 2007.

[33] Kai Liu, Yongchang Wang, Daniel L Lau, Qi Hao, et Laurence G Hassebrook. Dual-frequency pattern scheme for high-speed 3-d shape measurement. *Optics express*, 18(5):5229–5244, 2010.

[34] Nicolas Martin, Vincent Couture, et Sébastien Roy. Subpixel scanning invariant to indirect lighting using quadratic code length. Dans *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1441–1448. IEEE, 2013.

[35] Daniel Moreno, Fatih Calakli, et Gabriel Taubin. Unsynchronized structured light. *ACM Transactions on Graphics (TOG)*, 34(6):178, 2015.

[36] Shree K Nayar, Gurunandan Krishnan, Michael D Grossberg, et Ramesh Raskar. Fast separation of direct and global components of a scene using high frequency illumination. *ACM Transactions on Graphics (TOG)*, 25(3):935–944, 2006.

[37] Oline Vinter Olesen, Rasmus R Paulsen, Liselotte Hojgaard, Bjarne Roed, et Rasmus Larsen. Motion tracking for medical imaging: a nonvisible structured light tracking approach. *IEEE transactions on medical imaging*, 31(1):79–87, 2012.

[38] Tomislav Petković, Tomislav Pribanić, Matea Đonlić, et Nicola D'APUZZO. Software synchronization of projector and camera for structured light 3d body scanning. Dans *7th International Conference on 3D Body Scanning Technologies*, 2016.

[39] Jeffrey L Posdamer et MD Altschuler. Surface measurement by space-encoded projected beam systems. *Computer graphics and image processing*, 18(1):1–17, 1982.

[40] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, et Henry Fuchs. The office of the future: A unified approach to image-based modeling and spatially immersive displays. Dans *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 179–188. ACM, 1998.

[41] Szymon Rusinkiewicz, Olaf Hall-Holt, et Marc Levoy. Real-time 3d model acquisition. *ACM Transactions on Graphics (TOG)*, 21(3):438–446, 2002.

[42] Ryusuke Sagawa, Ryo Furukawa, et Hiroshi Kawasaki. Dense 3d reconstruction from high frame-rate video using a static grid pattern. *IEEE transactions on pattern analysis and machine intelligence*, 36(9):1733–1747, 2014.

[43] Joaquim Salvi, Xavier Armangué, et Joan Batlle. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern recognition*, 35(7):1617–1635, 2002.

[44] Joaquim Salvi, Joan Batlle, et E Mouaddib. A robust-coded pattern projection for dynamic 3d scene measurement. *Pattern Recognition Letters*, 19(11):1055–1065, 1998.

[45] Joaquim Salvi, Sergio Fernandez, Tomislav Pribanic, et Xavier Llado. A state of the art in structured light patterns for surface profilometry. *Pattern recognition*, 43(8):2666–2680, 2010.

[46] Joaquim Salvi, Jordi Pagès, et Joan Batlle. Pattern codification strategies in structured light systems. *PATTERN RECOGNITION*, 37:827–849, 2004.

[47] Venugopal Srinivasan, Hsin-Chu Liu, et Maurice Halioua. Automated phase-measuring profilometry of 3-d diffuse objects. *Applied optics*, 23(18):3105–3108, 1984.

[48] Junhua Sun, Guangjun Zhang, Zhenzhong Wei, et Fuqiang Zhou. Large 3d free surface measurement using a mobile coded light-based stereo vision system. *Sensors and Actuators A: Physical*, 132(2):460–471, 2006.

[49] Joji Takei, Shingo Kagami, et Koichi Hashimoto. 3,000-fps 3-d shape measurement using a high-speed camera-projector system. Dans *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3211–3216. IEEE, 2007.

[50] Marcelo Bernardes Vieira, Luiz Velho, Asla Sa, et Paulo Cezar Carvalho. A camera-projector system for real-time 3d video. Dans *Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops(CVPRW)*, page 96. IEEE, 2005.

[51] Yongchang Wang, Kai Liu, Qi Hao, Daniel L Lau, et Laurence G Hassebrook. Period coded phase shifting strategy for real–time 3-d structured light illumination. *IEEE Transactions on Image Processing*, 20(11):3001–3013, 2011.

[52] Yonatan Wexler, Andrew W Fitzgibbon, et Andrew Zisserman. Learning epipolar geometry from image sequences. Dans *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–209. IEEE, 2003.

[53] Clarence Wust et David W Capson. Surface profile measurement using color fringe projection. *Machine Vision and Applications*, 4(3):193–203, 1991.

[54] Song Zhang et Peisen Huang. Dans *Conference on Computer Vision and Pattern Recognition Workshop(CVPRW)*, page 28. IEEE, 2004.

[55] Song Zhang, Daniel Van Der Weide, et James Oliver. Superfast phase-shifting method for 3-d shape measurement. *Optics express*, 18(9):9684–9689, 2010.

[56] Song Zhang et Shing-Tung Yau. High-speed three-dimensional shape measurement system using a modified two-plus-one phase-shifting algorithm. *Optical Engineering*, 46(11):113603, 2007.

[57] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 2000.

[58] Zhengyou Zhang. Microsoft kinect sensor and its effect. *IEEE multimedia*, 19(2):4–10, 2012.

[59] Hong Zhao, Wenyi Chen, et Yushan Tan. Phase-unwrapping algorithm for the measurement of three-dimensional object shapes. *Applied optics*, 33(20):4497–4500, 1994.