

ÉCOLE NATIONALE DES SCIENCES GÉOGRAPHIQUES Luc BERAUD Hugo BOULZE Hugo DE PAULIS Matthieu GOUGEON Adriano MARZEC



Ingénieurs 1^{ère} Année

QUALIFICATION D'UNE CAMÉRA TEMPS DE VOL



Instructeur : Antoine Pinte

Juillet et Août 2018

Table des matières

Introduction	l	3
I. Présentatio	on de la Caméra	4
A. Fich B. Prin	ne technique ncipe de Fonctionnement	4 4
II. Premiers I	Pas	5
A. Prei B. Cali C. Déte D. Mes	miers Tests ibration ermination de la Focale sure dans le Nuage de points	5 6 8 8
III. Modélisat	tions 3D	9
A. Exp B. Trai	oort des Fichiers Vidéos itement des Données	9 10
IV. Optimisat	tion et Limites	11
A. Lum B. Cou C. Mat	nière lleur ériaux et Distance	12 14 15
		10
v. Travaux a	Pierrerue	18
A. Lase B. Pho C. Acq	ptogrammétrie Smartphone juisition avec la Caméra ToF	18 20 20
VI. Conclusio	on	22
ANNEXE :	Tutoriel du logiciel Voxel Extraction et Traitement des nuages de points Position des Ciblettes Acquisition photogrammétrique par Smartphone Comparaison du scan Laser par rapport à l'acquisition de la caméra ToF Scripts Python	

Introduction

Le projet de fin de stage à Forcalquier a été consacré au test d'une caméra temps de vol (ou ToF pour « Time of Flight »). Le principe de ces caméras est de pouvoir reconstituer instantanément une scène sous forme de nuages de points 3D grâce à l'émission de lasers infrarouges. Le projet s'articule autour d'une caméra de ce type, la OPT8241, développée par Texas Instruments. L'objectif au cours des trois semaines de projet est de qualifier cette caméra en vue d'une utilisation en reconstruction 3D. Pour ce faire, il sera nécessaire de la comparer à différentes techniques de reconstitution 3D : lasergrammétrie, photogrammétrie classique (par le biais d'un appareil photo haut de gamme et d'un smartphone). Les limites de la caméra ainsi que les conditions optimales de prises de vues ont également été testées et mises en œuvres.

Nous remercions chaleureusement la municipalité de Pierrerue (04) pour l'accès à la chapelle de Pierrerue.

I. Présentation de la Caméra

A. <u>Fiche technique</u>

<u>Modèle</u> : OPT8241 <u>Constructeur</u> : Texas Instruments <u>Prix d'achat</u> : 596 €

Configuration requise :

- Microsoft Windows 7 / Ubuntu 14.04 64 bit
- 2 GB de RAM
- 500 MB d'espace disque

Composants et caractéristiques techniques :

- 4 émetteurs lasers proches infrarouges (850nm)
- 1 récepteur optique (avec lentille)
- Taille du boîtier : 60 mm (W) x 48.5 mm (L) x 12.1 mm (H)
- Emprise d'illumination : circulaire (conique si en 3D)
- Résolution du capteur : 320 × 240 pixels
- Pixel Pitch (écart entre photosites) : 15 μm
- Fréquence d'acquisition : jusqu'à 150 images/seconde
- Champ angulaire (en degré): 74.4° (Horiz.) × 59.3° (Vert.)
- Un pas de vis pour trépieds d'appareil photo



Logiciel : La suite Voxel est indispensable pour observer, acquérir et traiter les données de la caméra.

Une documentation complète ainsi que les logiciels utiles à l'utilisation de la caméra sont présents sur le site de Texas Instruments au lien suivant : : <u>http://www.ti.com/tool/opt8241-cdk-evm</u>

B. Principe de Fonctionnement

La caméra ToF permet d'obtenir une image en 2,5D, c'est-à-dire une image en 2 dimensions mais contenant pour chaque pixel une information sur la distance. Ainsi, il est possible d'obtenir une représentation en 3 dimensions de la prise de vue. Contrairement à un appareil photographique dit « classique » qui ne fait que capter l'information lumineuse ambiante (on parle de capteur passif), la caméra ToF émet elle-même un rayonnement lumineux en proche infrarouge (environ 850nm) qui est réfléchi et qui est ensuite capté par la caméra (on parle de capteur actif). Elle présente ainsi l'avantage de fonctionner dans des endroits peu éclairés, ce qui est un atout considérable pour certains sites archéologiques ou historiques dépourvus d'éclairage.

L'intérêt de cette caméra est qu'elle peut obtenir de manière quasi-instantanée une information sur la distance. Pour cela, le signal est « pulsé » à une certaine fréquence. Ainsi la différence de phase entre le signal émit et reçu donne la distance parcourue et donc la distance à laquelle il a été réfléchi : c'est-à-dire la distance d'une surface matérielle. Une information en intensité lumineuse reçue (amplitude) dépend également de la distance de l'objet réfléchissant et de sa réflectance.



Le principe de reconstruction 3D de cette caméra est totalement différent de la vision stéréoscopique (comme la vision humaine, avec deux yeux qui, par combinaison, nous font percevoir le relief). Le calcul de la profondeur est relativement direct, là où par stéréoscopie il est nécessaire de résoudre des formules d'images, ainsi que d'avoir plusieurs prises de vues (contre une seule très rapide pour la ToF). Ces atouts peuvent apporter une solution efficace et dynamique pour des applications diverses telles que la reconnaissance gestuelle, la détection d'obstacles...

Figure 1 – Emission/Réception Texas Instrument |Time-of-Flight Camera – An Introduction





Figure 2 – Mise en évidence du faisceau circulaire infrarouge de la caméra. A gauche la caméra est éteinte, à droite elle est allumée.

II. <u>Premiers Pas</u>

A. <u>Premiers Tests</u>

Lors des premières acquisitions réalisées avec la caméra ToF, nous avons immédiatement constaté que la totalité des surfaces plates apparaissent courbées sur la prise de vue. Deux calibrations présentes par défaut dans le logiciel Voxel Explorer ont été utilisées : LongRange et ShortRange. Pour remédier à ce problème de nouveaux profils de calibrations ont été créés manuellement. Certains paramètres ne peuvent être calibrés précisément sans un matériel de précision et des conditions optimales.



Figure 3 – Exemple d'un mur supposé plat, vu de dessus.

De nombreux bugs ont été constatés, notamment lors de l'export des amplitudes. Il est parfois nécessaire de refaire la manipulation de nombreuses fois ce qui rend la tâche pénible et fastidieuse.





Figure 4 – Quelques bugs

Afin de contourner ce problème, il a paru plus intéressant de filmer une courte vidéo de la scène voulue et d'en extraire des frames, ces dernières ne présentant pas le problème présenté ci-dessus. De nombreux plantage du logiciel Voxel sont également de la partie, la plupart du temps lors des exports d'images ou des changements de calibration.

B. Calibration

Calibration effectuée	But	Procédé
Lens calibration	Corrige les distorsions optiques induites par la lentille. Calcule les paramètres : - coefficients de distorsion radiale - coefficient de distorsion tangentielle - coordonnées du PPA - distance focale	8 photos sur un damier
Common Phase Calibration	Met en corrélation la phase mesurée avec la profondeur réelle en un seul point (seul le pixel central est corrélé).	Mesure la phase sur un mur plat
Pixel-Wise Calibration	Corrige les variations de distance entre les pixels. Pour des applications de haute précision.	Pointer la caméra sur un mur plat où le mur recouvre tout le champ de vision tout en gardant l'appareil immobile

Nous nous sommes concentrés sur les calibrations que nous avons compris et celles qui ne nécessitaient aucun matériel particulier. Elles sont présentées dans le tableau ci-dessous :

La calibration retenue a été réalisée à 63,5 cm d'un mur peint supposé droit et a été faite pour le Common Phase Calibration et le Pixel-Wise Calibration. La Lens Calibration n'a pas été faite car pour toutes les calibrations où elle a été effectuée la géométrie des objets n'était pas correcte.

Common Phase Calibration Common phase offset computation. Please measure the p cx, cy	phase value at
From lens calibration, $cx = 159.0$, $cy = 119.0$	Calibrate
Distance from flat surface: 0,635 👻 m	
Number of frames to capture: 500 🚔	
Phase	
Threshold:	
	0%
Computing phase offset for second modulation frequency	
Common Phase Parameters	
tillum_calib = 60, tsensor_calib = 59,	
$phase_corr_1 = 872$, $phase_corr_2 = 823$,	

Figure 5 – Calibration

Grâce à cette nouvelle calibration, le phénomène de distorsion est corrigé : les surfaces gardent leur vraie géométrie.





Figure 6 – Mur de la Figure 3 après calibration

Une autre calibration a été réalisée dans les mêmes conditions mais à 1,635m d'un mur peint supposé lui aussi plat. Toutefois, le résultat est troublant car le mur obtenu après calibration est extrêmement courbé, même après plusieurs tests et tentatives de nouvelles calibrations identiques. Nous avons fait une autre acquisition de ce même mur mais avec la calibration réalisée précédemment et cette fois-ci le mur est parfaitement plat. Dans les deux cas le mur est situé à 1,635m de l'objectif et nous n'avons pas su expliquer ce qui différenciait nos deux calibrations pour donner de telles divergences entre les deux modèles.



Figure 7 – Mur avec 2 calibrations différentes



C. Détermination de la Focale

D'autres calibrations ont été réalisées dans le but de déterminer la focale de l'appareil. En effet, il était nécessaire de la déterminer en vue d'applications MicMac car les données exifs des images d'amplitude ne contenaient pas cette information. Cette calibration s'est déroulé sur un damier noir et blanc de 9 sur 13 où chaque carré mesure 4cm de côté.

La focale donnée par la calibration est de 228 pixels. Or la taille d'un pixel est de $15\mu m$ soit une focale de 3,42 mm.

Figure 8 – Calibration de la focale

D. <u>Mesure dans l'Image</u>

Pour tester la calibration, on effectue une mesure de distance sur un objet facilement mesurable dans la réalité. Les mesures sur le terrain ont été faites avec un distance mètre laser Leica DISTO.



Mesure CloudCompare : 0,977m



Mesure terrain : 0,997m



Mesure CloudCompare:0,336 m







Les mesures sur le terrain et celles effectuées sur le logiciel CloudCompare se situent dans le même ordre de grandeur mais celles-ci peuvent également varier en fonction de la calibration utilisée. Pour cette expérimentation, le nuage produit est métrique avec une certaine précision pour une calibration donnée. Une mauvaise calibration, un pointé approximatif sur CloudCompare (il n'y a pas toujours un point du nuage à la bordure précise de l'objet) ou une imprécision de mesure (matériel, opérateur) peuvent être à l'origine de ce problème.

IV. Modélisations 3D

L'export de nuages de points a permis de calibrer la caméra et d'effectuer toute une série de test pour comprendre son fonctionnement. Toutefois, la faible résolution des images obtenues ne permet pas l'utilisation d'un traitement photogrammétrique classique pour combiner plusieurs nuages de points. Pour résoudre ce problème, nous avons grandement augmenté le recouvrement des images pour que la reconnaissance de points de liaisons soit possible. L'utilisation du format de vidéo que propose la caméra est alors tout indiquée.

A. Export des Fichiers vidéos

Nous nous sommes rapidement rendu compte que le format vidéo (.vxl) utilisé par la caméra n'était pas directement exploitable. L'utilisation du logiciel Voxel Exporter, fourni par le fabriquant, permet d'extraire chaque frame de nuages de points. Ce logiciel fournit un fichier .bin, traité par un script python pour obtenir des fichiers .csv exploitables. Pour chaque frame on obtient un fichier contenant l'intensité de chaque pixel et un autre pour chaque coordonnée des axes XYZ.



Figure 10 – Organisation des fichiers .csv

Le script Python (modifié par la suite) permettant cette manipulation provient du site internet suivant : <u>https://e2e.ti.com/support/sensor/optical_sensors/f/989/t/503049</u>

Ces 4 fichiers sont concaténés par frame pour obtenir un unique fichier par frame plus facilement exploitable. Ce dernier, contenant les coordonnées de chaque pixel avec son intensité, peut être lu correctement par CloudCompare comme un nuage de points 3D. Reconstituer les images 2D de chaque frame est possible en utilisant seulement l'intensité.

B. Traitement des Données

Un résumé du traitement des données ainsi que les scripts Python sont présentés en Annexe.

Le nuage de points de chaque frame de la vidéo est extrait. Toutefois, chacune de ces frames possède son propre repère. Les nuages de points 3D de chaque frame ne sont donc pas superposables. Il est donc nécessaire de repositionner chaque nuage dans un même repère. Pour cela, deux méthodes ont été testées :

- réorientation direct des nuages de point 3D
- réorientation indirect par images 2D

La méthode directe consiste à réorienter directement les nuages de points 3D. En faisant des recherches sur Internet, nous avons pu vérifier qu'il existe bien des algorithmes capable de superposer deux nuages de points à partir de leur forme. Nous avons récupéré l'algorithme Fine Registration de CloudCompare. Cependant, les nuages de points étaient trop bruités ce qui entraînait parfois des résultats aberrants. En conséquence une méthode manuelle a été réalisée. Elle consiste à sélectionner au moins quatre paires de points sur des nuages de points pour les repositionner correctement. Cette méthode est longue, et donc réservée à un petit nombre de nuages de points. La précision obtenue (de l'ordre du centimètre) peut différer selon les opérateurs.





Figure 11 – Tachéomètre

La méthode indirecte consiste à réorienter des images 2D plutôt que des nuages de points 3D. Pour cela, les images créées précédemment à partir de l'intensité ont été utilisées. Le logiciel MicMac permet ensuite d'orienter chaque image dans un même système relatif. Cependant, les nuages de points sont déjà dans un système métrique. Pour transférer l'orientation et la translation de chaque image au nuage de points correspondant, il faut tout d'abord mettre la mise en place des images à l'échelle. Pour cela, deux méthodes existent :

- La mise à l'échelle simple : un même segment est pointé sur deux images différentes et la distance réelle correspondante est entrée. La commande Bascule de MicMac permet alors de recalculer les positions de chaque image dans un repère à l'échelle. Cette méthode, facile à mettre en place, peut être utilisée pour la reconstitution d'objets en 3D qui n'ont pas besoin d'être géoréférencés.
- Le repositionnement dans un repère : des ciblettes, dont les coordonnées connues dans un repère particulier, sont pointées. La commande GCPBascule recalcule alors les positions de ces images dans ce repère. Cette méthode permet de superposer des nuages issues de prises différentes ou par d'autres appareils. Toutefois, elle nécessite des ciblettes géoréférencées et un pointage rigoureux lors de la saisie des points d'appuis. La précision obtenue pour la position des ciblettes est d'environ 1,5 cm. Toutefois, à cause de la distorsion, la précision observée est plutôt de 3 cm.

Mise à l'échelle simple	Repositionnement dans un repère
mm3d setexif ".*png" F=3.42 F35=25 mm3d Tapioca Line ".*png" -1 10 mm3d Tapas RadialBasic ".*png" Out=MEP mm3d Apericloud ".*png" MEP Out=MEP.ply	mm3d setexif ".*png" F=3.42 F35=25 mm3d Tapioca Line ".*png" -1 10 mm3d Tapas RadialBasic ".*png" Out=MEP mm3d Apericloud ".*png" MEP Out=MEP.ply
mm3d SaisieBasc "(?.png !.png)" MEP SaisieBasc.xml mm3d SaisieMasq ?.png mm3d SaisieMasq !.png mm3d Bascule ".*png" MEP MEP_Echelle MesureIm=SaisieBasc-S2D.xml DistFS=0.28 *? et ! correspondent au nom des images	mm3d GCPConvert '#F=N_X_Y_Z' Appuis.txt mm3d SaisieAppuisInitQT "(? !).*png" MEP Appuis.xml Mesures.xml mm3d GCPBascule ".*png" MEP Bascule1 Appuis.xml mm3d SaisieAppuisPredicQT ".*png" Bascule1 Appuis.xml Mesures.xml mm3d GCPBascule ".*png" Bascule1 BasculeFin Appuis.xml
*? et ! correspondent au nom des images	mm3d GCPBascule ".*png" Bascule1 BasculeFin App Mesures-S2D.xml

Une fois la mise en place à l'échelle obtenue, un script python développé par nos soins permet d'extraire l'orientation et la translation de chaque image dans le dossier créé par MicMac. La matrice et le vecteur ainsi obtenus, chaque pixel de chaque nuage de points est repositionné grâce à la formule suivante :



Les nuages repositionnés peuvent alors être chargés dans CloudCompare, pour être fusionnés et nettoyés du bruit.

V. Optimisation et Limites

Il est important de connaître les limites de la caméra sur divers aspects (luminosité, portée, matériaux, etc.) en vue d'une utilisation en photogrammétrie. De ces limites, nous pouvons tirer des conditions optimales de prises de vues afin d'optimiser le rendu du nuage de points.

A. <u>Lumière</u>

1. Surexposition naturelle

Nous avons voulu mettre en évidence les limites des performances de la caméra en conditions lumineuses. Pour cela, plusieurs expérimentations ont été réalisées.

Dans un premier temps, la caméra est placée à l'ombre et vise un mur à l'ombre mais par un temps très ensoleillé. Il s'est avéré que le nuage de points contient alors de nombreuses anomalies (absences de points) rendant ainsi inutilisable le nuage 3D généré. Pour corriger cette perturbation, il a été nécessaire de couvrir la caméra.



Figure 12 – La caméra protégée par le parapluie

Pour mettre en évidence l'influence du soleil sur nos mesures, la caméra a été placée en plein soleil face un mur lui aussi exposé. Une courte vidéo montrant la réaction de la caméra à une alternance de nuages et de Soleil a été produite. La vidéo montre l'arrivée d'un nuage couvrant progressivement le Soleil.



Sous une légère couverture nuageuse

Figure 13 – Extrait de la vidéo du nuage 3D

Le surplus d'infrarouges apporté par la lumière du soleil est donc à l'origine de ces anomalies : les pixels sont saturés et ne donnent plus d'information. Ainsi, pour des applications en photogrammétrie, il est nécessaire d'effectuer les prises de vues par temps couvert (comme préconisé en photogrammétrie classique) voire de nuit où les infrarouges proches de ceux émis par la ToF sont quasi-absents. Cidessous, une autre expérience mettant en évidence ce phénomène. Ici, la saturation est locale et donc facilement identifiable.



Figure 14 – Visualisation de la saturation

Toutefois un pixel saturé apparaît dans le nuage de points 3D. En effet, la TOF créée un point très éloigné du sommet de prise de vue sur une surface concave. On remarque le même phénomène quand on réalise une acquisition en direction du ciel (pas de réflexion donc aucune donnée).



Figure 15 – Projection des points saturés ou non perçus

2. Auto-réflection

À courte distance, sur des matériaux réfléchissants ou des surfaces lisses, le récepteur sature à cause de ses propres LED, ce qui a pour effet de générer une zone inutilisable par manque d'information. Nous avons fait le test sur deux surfaces différentes :



Armoire métallique

Figure 16 – Réflexion

3. <u>Différence de sources</u>

Un test a été réalisé avec et sans éclairage artificiel afin de vérifier si ce dernier a un impact sur l'acquisition. Ici, une armoire métallique est éclairée par un néon : on ne constate aucune différence entre les deux acquisitions car le néon n'émet pas dans l'infrarouge. Nous supposons que d'autres types de lampes pourraient affecter la mesure, comme par exemple l'halogène qui émet fortement dans les infrarouges.





Figure 17 – Avec et sans lumière artificielle

B. <u>Couleur</u>

Quelques tests de colorimétrie ont été effectués et ont mis en évidence que la couleur d'un objet a peu d'influence sur sa capacité à réfléchir le proche infrarouge (PIR).

Ici, nous avons aligné 5 jerrycans de différentes couleurs, dans l'ordre qui suit : blanc, jaune, rouge, bleu, noir. Il faut souligner que le 5^e jerrycan (entouré en noir sur les images), est en fait un sceau en plastique noir. Aussi, le jerrycan jaune (pointé par une flèche jaune) n'est pas de même confection que les autres et semble réfléchir naturellement plus de rayons PIR que les autres.



Les réflections sont similaires indépendamment des couleurs



Il y a atténuation des nuages de points en fonction de la distance

La couleur noire est la seule exception à la règle puisque dès 1m, le sceau n'est plus identifiable. Nous remarquons la même particularité sur la prise de vue d'une roue de voiture ci-dessous.



Figure 19

Comme nous pouvons le voir sur cette prise de vue à 50cm, l'intensité des points associés au pneu de la voiture est bien plus faible que celle de tous les autres points (pneu pointé par la flèche noire). Cela est confirmé par le même contraste d'intensité au niveau des barrettes plastiques noires de la carrosserie (encerclées en rouge). Le phare avant (entouré en rose) semble avoir été traversé de part en part par les rayons PIR de la ToF, cela nous amène à considérer l'influence des matériaux sur la qualité d'une prise de vue ToF.

C. Matériaux et Distance

Pour tester la réflectivité en PIR des matériaux, nous avons décidé de faire une acquisition simultanée d'un large échantillonnage de matériaux. Ainsi, plusieurs objets sont disposés dans un espace sombre. Nous voulions faire varier la distance de prise de vue lors de l'acquisition ToF de ces objets. L'objectif était de comparer la portée maximale de prise de vue pour différents matériaux et d'en déduire leur réflectivité.



Les objets ont été disposés de manière à ne pas interagir entre eux lors de l'acquisition. Il fallait cependant que l'installation soit suffisamment compacte pour que nous puissions commencer à une courte distance, le champ d'acquisition étant fixe.



Nuage de points des matériaux Prise de vue à 2,06m



Nuage de points des matériaux avec des points grossis Prise de vue à 2,06m

Figure 20 – Test des matériaux

- Flèche marron = Table en bois
 Flèches noires = Table et chaise en métal
 Flèche verte = couverture textile
 Flèche jaune = jerrycan en plastique jaune
 Rond blanc = cible "Black and White" (BW) en papier

Les nuages de points ci-dessus ont servi de témoins pour nos tests de distance car tous les objets sont perçus par la ToF indépendamment des matériaux qui les constituent.

Notre expérience a été effectuée à proximité d'une voiture, ce qui explique l'apparition d'une bande réfléchissante (bande de points rouges) en haut à droite des images dès 2,47m de distance.



Témoin en haut à gauche(2,06m), 2,47m en haut à droite, 3,10m en bas à gauche, 3,97m en bas à droite

D'après ces images, 2,47m semble être la limite à laquelle tous les objets sont encore perceptibles et identifiables. Par la suite, les objets seront de moins en moins perceptibles.

Nous avons donc effectué une nouvelle prise de vue à 3,97m en changeant la position du textile afin de constater ou non une différence de réflectivité car nous avions des doutes sur l'effectivité de notre protocole. Enfin, nous avons pu établir une hiérarchie des matériaux :

Classement final des matériaux testé en fonction de leur réflectivité (du plus perceptible au moins perceptible) :





Prise de vue "Témoin" à 2,06m



Prise de vue "floue" à 4,4m

Figure 22

A partir de 4,4m, il n'y a plus d'objets discernables par la ToF mais nous pensions pouvoir aller plus loin et avons donc cherché à expliquer la divergence de nos points. Nous avons remarqué durant nos acquisitions du bruit spécifique et redondant sur toutes nos mesures, nous avons donc mis en évidence son influence sur la clarté de notre nuage de points.



Prise de vue à 2m, nuage de point orienté perpendiculairement à la prise de vue Le bruit est entouré en noir (cf légende plus haut pour les flèches)

Dès 2m, le bruit est important relativement à notre nuage de point exploitable. Cela s'explique par la présence de rayons solaires à travers l'ombre des feuilles. Le ToF est très sensible à la lumière naturelle. Ce phénomène s'exacerbe à mesure que l'on s'éloigne des objets ciblés.



Figure 24 – Mise en évidence du bruit à 2,47m (à gauche) et à 3,97m (à droite)

Un usage de la ToF en intérieur ou par temps nuageux est encore une fois préconisé. Malheureusement, nous ne pouvions effectuer nos tests de distance dans de telles conditions.

V. <u>Travaux à Pierrerue</u>

Afin de comparer les nuages 3D issus de différentes techniques, nous avons décidé de nous rendre sur le site de la chapelle St Pierre dans la commune de Pierrerue (04). Ce site a été choisi pour sa proximité et notre bonne connaissance du site car des travaux de photogrammétrie architecturale ont été réalisés quelques semaines auparavant.

A. Lasergrammétrie

Une acquisition par laser Leica P40 a été effectuée dans la chapelle St-Pierre à Pierrerue dans le but de comparer avec le rendu de la caméra ToF. Positionné au centre de la chapelle, le laser scanne à 360° l'intérieur du bâtiment avec une résolution de 1cm pour 10m afin d'obtenir un nuage de points global. Ensuite, un scan de la façade intérieure au niveau du chœur est réalisée avec une résolution plus élevée (5mm pour 10m) dans la zone qui nous intéresse. Il réalise aussi une acquisition photographique afin d'obtenir un nuage de points en couleurs réelles, ce qui permet de reconnaître facilement les détails.



Figure 25 – Acquisition par lasergrammétrie dans la chapelle

Nous avons placé des cibles de type « black and white » sur les murs (cf Annexe) qui correspondent à des points de repérage pour lier les photographies entre elles. Le laser scanne certaines cibles qui seront géoréférencées dans un repère commun afin d'orienter les photos issues de la caméra. Dans le logiciel Cyclone, les cibles identifiées sont considérées comme des vertex. Le logiciel récupère les coordonnées des cibles ce qui permettra ensuite de connaître les coordonnées de toutes les cibles géoreférencées dans ce repère locale.



Figure 27 – Nuage de points sur Cyclone



Figure 26 – Pointage des vertex

Pour visualiser les nuages points et réaliser des mesures, il est nécessaire d'utiliser CloudCompare (ou tout autre logiciel de visualisation 3D métrique). Cette application affiche l'ensemble des nuages de points dans un système métrique. Chaque coordonnée d'un point peut alors être connue dans le repère du laser.



Figure 28 – Image obtenue par lasergrammétrie sur CloudCompare

B. Photogrammétrie Smartphone

Une autre méthode d'acquisition photogrammétrique "low-cost" est la prise de vue avec un smartphone. Cela est possible avec tout smartphone classique. Nous avons effectué une acquisition avec un Hisense C30 Rock.

Le but étant d'essayer de comparer ces deux méthodes low-cost. L'acquisition en intérieur du smartphone a put être comparée à l'acquisition de la caméra ToF et à l'acquisition laser. La prise de vue en extérieur de la chapelle de Pierrerue a put être comparée à des modèles existants: ceux d'une acquisition photogrammétrique classique (appareil photo et laser).

Les écarts au nuage de point ont été en moyenne de 4,2cm comparé à une acquisition avec appareil photographique (extérieur seul) et de 5,1cm (extérieur) et 3,1cm (intérieur) comparé à une acquisition laser. L'écart moyen en intérieur entre la ToF et la prise smartphone est de 1,6cm. Les écarts sont donc plus élevés que ceux obtenus avec la caméra ToF, bien que sur des surfaces plus élevées et avec une reconstruction moins bien réalisée.



Des détails et captures sont disponibles dans l'annexe.

Figure 29 - Superposition des nuages de la caméra Tof et acquisition smartphone

C. Acquisition avec la Caméra ToF

Pour modéliser les murs dans le chœur de la chapelle, nous avons enregistré des vidéos par la caméra afin de récupérer des images du mur pour les traiter sur MicMac. Ce procédé permet d'atteindre les recoins comme les alcôves. Il suffit de rassembler les nuages de points de plusieurs vidéos pour représenter un modèle unique.



Figure 30 – Modélisation du chœur à partir de la ToF et superposition sur le scan laser



Le nuage de points obtenu par les captures de la caméra temps de vol peut-être superposé sur le scan du laser grâce au géoréférencement des nuages de points dans le même repère. L'avantage de cette technique est d'obtenir un modèle 3D d'une zone en entier, en particulier les détails non visibles par le laser comme les alcôves ainsi que le mur derrière l'autel. La modélisation par la caméra permet donc de compléter le nuage de points du laser qui est plus précis.

Figure 31 – Écart de distance du scan laser par rapport au nuage de points de la caméra (cf. Annexe)





Figure 32 – Histogramme des écarts entre les deux modèles

Le résultat est satisfaisant pour obtenir un modèle 3D détaillé puisque que l'écart moyen est de 4,2mm entre le scan laser et le nuage de points issu de la caméra après nettoyage.

VI. <u>Conclusion</u>

L'OPT8241 et plus globalement les prises de vue Time of Flight semblent pouvoir se faire une place dans des travaux de reconstruction 3D de précision. Le rapport qualité/prix est indéniablement en faveur de ce prototype comparé au prix du laser P40 utilisé dans notre projet.

Cependant, là où la caméra ToF brille par sa simplicité d'utilisation, sa rapidité d'acquisition de données denses (coordonnées 3D, intensité du signal) et par ses dimensions compactes, elle pêche sur l'étendue de ses cas d'utilisations. En effet, il a été mis au jour que la luminosité naturelle, la distance à l'environnement étudié et les matériaux le composant (quand ils ne sont pas simplement noirs) sont autant de facteurs très limitants pour l'utilisation de cette caméra.

Ce type d'acquisition n'est donc pas révolutionnaire dans les opportunités qu'elle donne à la reconstruction 3D. Comme pour les autres techniques, elle souffre de défauts que l'ingénieur doit considérer lors de ses travaux. Ainsi, la caméra ToF est complémentaire des autres techniques d'acquisition 3D, comme prouvée dans les travaux à Pierrerue. Son utilisation conjuguée à une lasergrammétrie classique complète l'information de cette dernière.

Après ces deux semaines d'études du système ToF, et plus particulièrement du prototype de la caméra OPT8241, nous pouvons produire quelques pistes d'amélioration du produit testé. La résolution de ce prototype a semblé bien insuffisante pour des travaux de précision. Cependant, l'augmentation de celle-ci résulterait en une augmentation du prix de l'appareil, c'est donc un compromis à trouver entre l'utilisation à faire de cet appareil et son prix.

Apprentissage personnel

Travailler sur un prototype de caméra, qui plus est sur un système d'acquisition émergent dans le domaine de la photogrammétrie, fût une expérience motivante bien que fluctuante dans l'enrichissement qu'elle pouvait nous apporter. En effet, un prototype ne demande qu'à être découvert et étudié mais il reste un prototype, son fonctionnement peut être instable, et ce fût particulièrement le cas avec l'OPT8241.

En dehors de ces impondérables récurrents, nous avons apprécié la liberté d'expérimentation que nous offrait ce projet. Mettre au point des protocoles, émettre des hypothèse, les réfuter et les confirmer, furent autant d'activités orchestrées de notre chef et agréables à accomplir.

Encore une fois, nous avons appris l'importance d'un travail d'équipe coordonné et d'un entretien régulier de la documentation commune. Nous avons plaisir à voir que le travail de chacun aboutit à un rendu que nous estimions dense et riche d'informations pour les instituts de géomatique désirant acquérir ce type d'appareil.

ANNEXE : Tutoriel du logiciel Voxel

La caméra Time-of-Flight est utilisable via le logiciel Voxel développé par Texas Instrument (le fabricant de la caméra). Celui-ci se décompose en plusieurs branches : Voxel Viewer, Voxel Exporter et Voxel Programmer.



Tout d'abord, Texas Instrument met à disposition des outils de développement logiciel (ou SDK) afin d'exploiter soi-même au mieux le produit sans passer par les outils fabricants qui, bien que en facilitant l'usage restreint les possibilités de la caméra.

Dans notre projet, nous nous sommes concentré sur la version fabricant car celle ci satisfaisait amplement la direction que nous souhaitions donner à notre projet. Nous n'aborderons donc pas en détail les outils SDK et Voxel Programmer dans cette annexe.

I. Voxel Viewer (v0.9.10 Windows)

Voxel Viewer (VV) est l'interface PC de la caméra. Lorsque la caméra est connectée via USB à l'ordinateur où cet outils est lancé, il permet d'avoir un retour image en direct de la caméra, de modifier les options de prise de vue et de faire des acquisitions.



<u>Utilisation</u>

Voxel Viewer a de l'intérêt principalement caméra connectée (mis à part consultation de calibration ainsi que les exportations que fait Voxel Exporter en mieux).

Au lancement de VV (Voxel Viewer pour la suite), la connexion au flux vidéo de la caméra se fait automatiquement si celle-ci est sous tension, allumée et connectée au PC. Il est théoriquement possible de la connecter après lancement via [file -> Connect Depth Camera].

Difficultés de lancement :

Nous avons eu de nombreux problèmes pour correctement établir le flux video entre la caméra et le PC. Outre le fait que le démarrage du flux était quasiment impossible sur certains PC, il fallait souvent relancer VV caméra allumée et brancher et espérer la connexion automatique. Sélectionner une calibration par défaut dans [Camera Profiles] par exemple Short Range, fermer le logiciel puis le relancer peut fonctionner (cela doit aider l'initialisation).

Parfois, démarrer l'échange de flux est très peu fréquent malgrés de multiples essais. Il peut s'avérer utile (bien qu'étrange) de devoir correctement lancer le flux sur un PC « compatible à l'humeur » de la caméra puis une fois le flux établi, sans éteindre la caméra, rebrancher le port USB sur le PC voulu (et ensuite lancer VV). Peut s'avérer très délicat pour aller sur le terrain, néanmoins ne pas perdre espoir : certains PC portables ont de bonnes affinités avec la TOF.

Modifier les paramètres (notamment la calibration) peut faire « freezer » la caméra : le flux s'arrête.

Présentation de l'interface(cf. la capture légendée de l'interface qui suit) :



Ecran général de Vowel Viewer, caméra connectée.

1) Le nuage de point 3D en temps réel. On peut grossièrement naviguer dans la vue 3D (seul zoom/ rotation autour du « centre optique »). On peut définir avec les curseurs en dessous le « Depth threshold » : seuil de profondeur de l'image affichée (et non enregistrée).

2) Fenêtre d'amplitude: image de l'amplitude reçue par le capteur. On peut définir avec les curseurs en dessous le « Threshold » : seuil d'amplitude de l'image affichée

3) Fenêtre de phase: image caractérisant la profondeur de la vue en cours (ou la distance à la caméra pour chaque point).

4) Ce sont ici trois onglets qui cohabitent : *Watch List, Data Flow Diagram, Logs*

4.1) Watch List: diverses caractéristiques concernant l'acquisition

Watch List												₽×
Point	Phase (Avg, Std)	Amplitude (Avg, Std	Ambient	Flags	Z, meters (Avg, Std)		Spatial Statisti	ics		Obser	ved Parameters	
						From: 0 🚖	x 0 🌲 to:	319 🌲 🗴 23	9 ≑	Name	Value	<u>^</u>
							Average	Std	~	comp_fb_error_cnt	0	
						amplitude_avg	29.08	32.23		tsensor	51	
						nhace avo	1113.15	162.24	-		10	
Watch List Data	Flow Diagram Logs	;										
Device Connected	👿 Temporal Statis	stics			@(59, 0) -> Amp	litude: 0027, 1	Phase: 0809,	Ambient: 06,	, Flag	s: 00, Z: 01.6353m	Texas Ins	TRUMENTS

4.2) Data Flow Diagram

Data Flow Diagram								
	Raw Frame (unprocessed)	Raw Frame (unprocessed)	Raw Frame (processed)	Raw Frame (processed)	Depth Frame	Depth Frame	Point Cloud Frame	Point Cloud Frame
		1]]		1
Watch List Data Flow Diagram Logs								

Il permet de visualiser les étapes de traitement du signal de l'acquisition brute jusqu'à la création du nuage de point. On peut ajouter des filtres intermédiaires prédéfinis via [clic droit] sur les cases puis [add filter...]. Un double clic sur les filtres rajoutés permet d'ajuster leurs paramètres. Des conseils et explications sur les filtres sont détaillés dans la documentation constructeur : par exemple l'application d'un filtre médian puis temporel permet une réduction du bruit.

4.3) *Logs :* Permet un diagnostique infaillible sur le fonctionnement ou non du flux vidéo. Il est possible de l'exporter.

Logs		5 ×
Default Log Level: Information 💌	Log file: C:\Users\formation\.Voxel\logs\log.txt	Apply
Voxel Viewer v0.9.10 Texas Instruments Inc.		▲ □
INFO: ParameterDMLParser: Found register map with name 'OPT9221' (VPG 1P0) INFO: HardwareSenialzer: Received 45565 bytes from hardware in 2.0632 s INFO: DtazDCodec: Original number of 8-bit offsets = 3030		
Watch List Data Flow Diagram Logs		

5) Ici il y a deux onglets : *Frequently used Parameter, Parameters.*

5.1) Frequently used Parameter: permet l'affichage et le réglage de divers paramètres classiques et importants : Intensité lumineuse émise (augmente le signal émis et donc de même le signal reçu), période cyclique d'acquisition du signal (analogue au temps de pose d'un APN classique). Il y a aussi les images par seconde (FPS), la distance «de confiance» et la taille du capteur utilisée. Modifier ces paramètres peut influer sur la calibration en cours (cf. DOC constructeur, l'intensité a une influence limitée, là où l'integration duty cycle a un impact important).

5.2) Parameters: permet l'affichage et le réglage de divers paramètres fins, nous n'y avons que superficiellement touché car les paramètres de *Frequently used Parameters* suffisent à conduire une étude détaillé de l'appareil.



6) Barre d'enregistrement vidéo: permet d'enregistrer une vidéo (1 image (frame) = 1 nuage de point acquis). L'arrêt propose l'enregistrement en format .vxl (la conversion sera abordée plus loin).

Intéressons nous à présent aux onglets outils en haut à gauche de VoxelViewer :

- [File -> Open file stream] Ouvrir une vidéo capturée.
- [File -> Export Raw stream] Exporter la vidéo en fichier .bin (lui aussi à convertir, le détail est disponible dans la partie dédiée au traitement des acquisitions). Attention, l'export peut faire crasher VV, utilisez de préférence Voxel Exporter.
- [Camera Profiles] Onglet utilisé couramment, il permet la sélection de différentes calibrations par défaut (bien plus fiable pour le fonctionnement de VV mais avec souvent de très grosses déformations). L'outil Calibration Wizard permet de réaliser une calibration soi-même de divers

paramètres (lentille, phase, etc) et ainsi avoir un environnement mesurable et cohérent. Se référer à la partie calibration du rapport de projet pour plus de détails.

- [Settings -> Parameters] Divers paramètres
- [Windows] Paramètre l'affichage des onglets et fenêtres.

II. Voxel Exporter (Windows)

Voxel Exporter (VE) est l'outil d'export/extraction des vidéos. Ceci est possible sous Voxel Viewer mais le logiciel crash régulièrement pendant l'export et est moins pratique pour convertir plusieurs fichiers à la suite.

Data Type: Point Cloud	-
D:/P4_CameraTempsDeVol_tmp/video_table.vxl	
D:/P4_CameraTempsDeVol_tmp/PointCloud.bin	
	0%

Fenêtre de Voxel Exporter

VE exige le choix du type de donnée à extraire de la vidéo au format .vxl (Ambient, Amplitude, Point Cloud, Phase, etc.)

[capture paramètres]

Ensuite il faut choisir le fichier .vxl à convertir puis le fichier de sortie qui sera au format .bin (illisible directement, se référer à la partie sur le traitement des données pour la solution que nous avons utilisée).

A priori, il n'y a pas de méthode pour extraire plusieurs types de données en une fois (nuage de point, amplitude ou autre).



ANNEXE : Extraction et Traitement des nuages de points

ANNEXE : Position des ciblettes



ANNEXE : Acquisition photogrammétrique par Smartphone

Deux acquisitions par smartphones ont été menées à la chapelle de Pierrerue: une en extérieur et une en intérieur.

Ce choix a été fait pour avoir une acquisition en milieu favorable (lumineux) à savoir en extérieur, et une en environnement défavorable (sombre) : à l'intérieur.

Le but de cette acquisition est de qualifier cette méthode de prise de vue photogrammétrique lowcost, ainsi que de la comparer avec d'autres méthodes «classiques» à savoir station Laser (ici Leica P40) et appareil photographique (ici Canon 70D ou 80D). Le smartphone utilisé était le C30 Rock de Hisense (nous n'avons pas pu trouver d'information sur la taille du capteur ou des photosites ni de la focale équivalente, contrairement à d'autres smartphones).

Logiciel de capture: Open Camera (permet de fixer la mise au point)

Logiciels de traitement: MicMac et Cloud Compare

Il n'était pas possible de contrôler ni le temps de pose ni l'ouverture (probablement fixe pour ce type de capteur), la focale non plus (fixe aussi). Nous avons essayé une prise avec mise au point fixe : mise au point fixée à l'infini inutilisable, mise au point verrouillée sur une distance peu stable (dérive et faible profondeur de champ : nous l'avons libérée plusieurs fois pour l'ajuster).

Le fait de ne pouvoir avoir de plus grande focale optique nous obligeait à être relativement proche donc avoir des différences de plans de mise au point très variables.

Prise de vue en intérieur:

<u>Cible</u> : le chœur, façade intérieur principale (Est)

<u>Conditions</u> : prise de vue en éclairage artificiel faible.

Pour la calibration, nous avons réutilisé celle faite en extérieur, celle en intérieur étant inexploitable.

<u>Principal problème rencontré</u> : flou de mouvement. Nous ne pouvions modifier le temps de pose et monter en ISO mal gérée par l'appareil. De plus, la prise de vue s'est faite sans trépied. La mise au point était peu fiable (dérive et faible profondeur de champ : nous l'avons libérée plusieurs fois pour l'ajuster).

<u>Traitement</u> <u>Modèle de calibration</u> : Radial standard

Précision : sub-centimétrique.



Superposition de l'acquisition smartphone et caméra ToF

La mise en place des clichés est correcte. Cependant, nous remarquons des zones vides ou non homogènes qui ont été difficilement reconstruites.



Écart acquisition ToF-Smartphone



Écart prise de vue laser - Smartphone

Prise de vue en extérieur:

<u>Cible</u> : façade extérieure principale (Ouest)

<u>Conditions</u> : prise de vue en éclairage naturel, ombragé.

<u>Principal problème rencontré</u> : La mise au point était peu fiable (dérive et faible profondeur de champ : nous l'avons libérée plusieurs fois pour l'ajuster).

Traitements

Modèle de calibration : Radial standard

Précision : sub-centimétrique.

Résultat de la corrélation dense MicMac



La mise en place est correcte. La mise en référence dans le repère local de la chapelle nous a permis de comparer cette modélisation avec celle acquise par laser et appareil photographique.

Les captures suivantes montrent les écarts entre les nuages de points spécifiés (en mètre) avec la légende accolé à l'histogramme.



Résultat de la corrélation dense MicMac

Écart entre la prise via appareil photographique du groupe 1 (semaine architecture) et acquisition smartphone

Résultat de la corrélation dense MicMac



Écart entre la prise via Laser du groupe3 (semaine architecture) et acquisition smartphone

Nous remarquons que la modélisation par smartphone est relativement proche des méthodes classiques et établies.

Surtout, cet écart est très proche de celui mesuré dans les nuages de points en intérieur de la chapelle entre la caméra temps de vol et une autre acquisition laser (environ 4mm).

<u>Conclusion</u>

Les écarts au nuage de point ont été en moyenne de 4,2cm comparé à une acquisition avec appareil photographique (extérieur seul) et de 5,1cm (extérieur) et 3,1cm (intérieur) comparé à une acquisition laser. L'écart moyen en intérieur entre la ToF et la prise smartphone est de 1,6cm. Les écarts sont donc plus élevés que ceux obtenus avec la caméra ToF, bien que sur des surfaces plus élevées et avec une reconstruction moins bien réalisée.

Nous pouvons en conclure que la qualité esthétique d'une acquisition par smartphone est moindre (colorimétrie, netteté, etc.), cependant le traitement des images a permit de corriger efficacement les déformation géométriques plus importantes (d'après les fichiers de calibration) et ainsi obtenir une qualité géométrique du nuage de point comparable à une acquisition classique.

Une acquisition par smartphone semble comparable en résultat (nuage de point) à une acquisition avec la caméra temps de vol. L'environnement pose problème aux 2 méthodes mais dans des conditions opposées (luminosité). Le rendu graphique est meilleur pour le smartphone de même que le posttraitement plus simple, cependant l'acquisition demande plus de rigueur et de temps. Une acquisition par smartphone «à la volée» est bien plus économique et légère qu'une acquisition photogrammétrique «classique» pour un résultat géométrique comparable. En revanche, l'aspect esthétique est plus défavorable pour une acquisition smartphone avec notre prise de vue. Cependant, les capteurs et logiciels de calculs photographiques embarqués dans les smartphones étant très différents les uns des autres, il est probable que certains appareils soient plus performants dans le domaine (même bas de gamme). Il reste cependant difficile de gagner en résolution étant donné que les valeurs de distance focales sont courtes et souvent fixes (zoom numérique bien souvent seul). Cela est handicapant pour une combinaison de distance à la cible et de profondeur de champ qui doit être relativement grande. Au vu des progressions dans le domaine de la photographie smartphone, il est très probable que de grands progrès comblent en partie ces lacunes (colorimétrie et focale).

ANNEXE : Comparaison du scan Laser par rapport à l'acquisition de la caméra ToF

Toutes les distances sont en mètres.









Vue côté droit (Sud)

<u>ANNEXE</u> : Script Python – Extraction du fichier .bin et création des images et nuages de points

```
import numpy as np
from skimage import io
import glob
import os
def coordonnees(num_img):
     Va chercher les coordonnées x, y, z, i des points dans les images en parcourant
    mat_i = np.genfromtxt('PointCloud_bin/frame' + num_img + '_i.csv', delimiter=","
mat_x = np.genfromtxt('PointCloud_bin/frame' + num_img + '_x.csv', delimiter=","
mat_y = np.genfromtxt('PointCloud_bin/frame' + num_img + '_y.csv', delimiter=","
mat_z = np.genfromtxt('PointCloud_bin/frame' + num_img + '_z.csv', delimiter=","
     n,p = np.shape(mat_i)
     i, x, y, z = [], [], [], []
     for k in range(n):
          for l in range(p):
                i.append(mat_i[k,l])
                x.append(mat_x[k,l])
                y.append(mat y[k,l])
                z.append(mat_z[k,l])
     return i, x, y, z
def fichier_cor(num_img):
     Crée le fichier de toutes les coordonnées des points par image
     i, x, y, z = coordonnees(num img)
     fichier = open('Clouds/coordonnees_frame' + num_img + ".txt", "a")
     for j in range(len(i)):
           if np.sqrt(x[j]**2+y[j]**2+z[j]**2) < 3 and np.sqrt(x[j]**2+y[j]**2+z[j]**2)
                fichier.write(str(x[j]) + ',' + str(y[j]) + ',' + str(z[j]) + ',' + str(
     fichier.close()
if __name__ == "__main__":
     print("\nI. Conversion des donnees en .csv :")
     fileList=['PointCloud']
     fileDetails=dict()
     for fileT in fileList:
          fileDetails[fileT]=dict()
    fileDetails['PointCloud']['dtype']=np.float32
fileDetails['PointCloud']['nQuantities']=4
fileDetails['PointCloud']['quantityNames']=np.array(['x','y','z','i'])
fileDetails['PointCloud']['reverseRowCols']=True
     fileSizeQ=np.product((240,320))
     for fileT in fileList:
       fileName=fileT+".bin'
                                                                                        # Adds .bin to the
```

```
dtype=fileDetails[fileT]['dtype']
nQuantities=fileDetails[fileT]['nQuantities']
                                                               # datatype defined
                                                      # Number of quantities p
 quantityName=fileDetails[fileT]['quantityNames'] # This is used to just name (
 filePath=fileName
 if(os.path.exists(filePath)):
                                                               # Checking if file
       data=np.fromfile(filePath,dtype=dtype)
                                                     # Loading Raw binary file ;
       if(np.size(data)%(fileSizeO*nQuantities)==0):
                                                       # Checking if file size
            nFrames=np.size(data)/(fileSizeQ*nQuantities)# Calculating number of
            print ("INFO: %04d Frames in file [%s]"%(nFrames,filePath))
       else:
            print ("ERROR: File size [%d] for file [%s] not a multiple of frame
 else:
       print ("ERROR: Unable to open file [%s]"%(filePath))
 csvDir=fileName.replace('.',' ')
                                              # Directory where CSV files will t
 if(not os.path.exists(csvDir)):
                                                           # Checking for existan
                                                                 # Creation of di
       os.makedirs(csvDir)
 if(fileDetails[fileT]['reverseRowCols']):
       data=data.reshape(nFrames, 240, 320, nQuantities).swapaxes(-1,-2).swapaxes(-
 else:
       data=data.reshape(nFrames, 320, 240, nQuantities).swapaxes(-1,-3)
 for cθ in np.arange(nFrames):
                                                               # Loop for each fre
       for c1 in np.arange(nQuantities):
                                                               # Loop for each que
            dataSave=data[c0,c1]
                                                                # Extracted data :
            csvFileName=csvDir+'/frame%04d %s.csv'%(c0,quantityName[c1])
            np.savetxt(csvFileName,dataSave,delimiter=',',newline='\n') # Saving
print("
          -> Conversion terminee")
```

```
print("\nII. Reconstruction des nuages de points et des images :")
```

```
# Creation du dossier de nuages de points :
try:
   os.mkdir("Clouds")
except OSError:
    pass
# Creation du dossier des images d'amplitude :
try:
   os.mkdir("Images")
except OSError:
    pass
print("
          1. Reconstruction des nuages de points :")
nbre_fic = len(os.listdir("PointCloud bin"))//4
n = \theta
for i in range(nbre_fic):
    if i % 2 == 0 :
        if i<10:
            text = fichier_cor('000' + str(i))
        elif i<100 :
            text = fichier_cor('00' + str(i))
        elif i<1000 :
            text = fichier_cor('0' + str(i))
        else :
```

```
text = fichier_cor(str(i))
```

```
print(" -> Reconstruction terminee")
print(" 2. Reconstruction des images :")
list_csv = np.sort(glob.glob("PointCloud_bin/*_i.csv"))
i = 0
for csv in list_csv :
    if i % 2 == 0 :
        ing = (np.genfromtxt(csv,delimiter=','))
        ing=((ing-np.min(ing))/(np.max(img)-np.min(ing))*255).astype(int)
        io.imsave("Images/img_"+"%04d"%(i)+".png",img)
    i += 1
print(" -> Reconstruction terminee")
```

<u>ANNEXE</u> : Script Python – Extraction de l'orientation des images et repositionnement des nuages de points

```
import numpy as np
import os
def extraire_position(nonfic) :
     fic = open(nonfic,"r")
     lines = fic.readlines()
     fic.close()
     for i in range(len(lines)) :
         lines[i] = lines[i].split("\n")
    for line in lines :
    if "<Centre>" in line[0] :
         vect = line[0].split(" ")[15:]
if "<L1>" in line[0] :
    L1 = line[0].split(" ")[25:]
         if "<L2>" in line[0] :
              L2 = line[0].split(" ")[25:]
         if "<L3>" in line[0] :
              L3 = line[0].split(" ")[25:]
    mat_rot = np.zeros((3,3),float)
    mat_rot[0,0] = float(L1[0][4:])
    mat rot[0,1] = float(L1[1])
mat_rot[0,2] = float(L1[2][:len(L1[2])-5])
mat_rot[1,0] = float(L2[0][4:])
    mat rot[1,1] = float(L2[1])
    mat_rot[1,2] = float(L2[2][:len(L2[2])-5])
mat_rot[2,0] = float(L3[0][4:])
    mat rot[2,1] = float(L3[1])
    mat_rot[2,2] = float(L3[2][:len(L3[2])-5])
    vect_tran = np.zeros((3),float)
    vect_tran[0] = float(vect[0][8:])
    vect tran[1] = float(vect[1])
    vect_tran[2] = float(vect[2][:len(vect[2])-9])
    return mat_rot, vect_tran
os.mkdir("Nuages_orientes")
list_ori = np.sort(os.listdir("Images/Ori-BasculeFin"))
list ori = list ori[2:len(list ori)-2]
list_nua = np.sort(os.listdir("Clouds"))
lon = len(list_ori)
for i in range(lon) : #len(list_ori)
    nuage = np.genfromtxt("Clouds/"+list_nua[i],delimiter=',')
    mat_rot, vect_tran = extraire_position("Images/Ori-BasculeFin/"+list_ori[i])
fic = open("Nuages_orientes/nua_ori_"+"%04d"%(i)+".txt","w")
#print(str(i)+"/"+str(len(list_ori)))
     for j in range(len(nuage)) : #len(nuage)
         coord i = nuage[j][:3]
         coord_r = mat_rot.dot(coord_i)+vect_tran
         fic.write(str(coord_r[0])+","+str(coord_r[1])+","+str(coord_r[2])+","+str(nu
     fic.close()
```