

# Appariement 3D d'objets archéologiques

**Adrien Bosson**

**Juillet 2022**

Domaine Ingénierie et Architecture

Master conjoint UNIGE-HES-SO en développement territorial

Orientation Ingénierie Géomatique parcours brevet

Directeur : Dr Adrien Gressin

Expert : Dr Matthieu Deveau

Mandant : Get in Situ / Xavier Muth

Mémoire n° : 1020



**UNIVERSITÉ  
DE GENÈVE**

**Hes·so**  
Haute Ecole Spécialisée  
de Suisse occidentale

## Résumé

Les remontages en archéologie lithiques sont des outils précieux permettant de déduire des informations sur les sites archéologiques qui permettent de mieux connaître les civilisations qui y ont vécu des dizaines de milliers, voir centaines de milliers d'années avant nous. Aujourd'hui, ces remontages se font principalement manuellement. Il s'agit de travaux long et fastidieux qui nécessitent du personnel spécifique. C'est dans ce cadre que s'inscrit le présent travail avec comme but de développer une méthode d'appariement numérique automatisée.

La méthodologie présentée dans ce travail propose différentes techniques d'appariement pouvant permettre le remontage. Les calages de nuages à l'aide de points d'intérêts 3D et les calages d'images 2D des faces sont comparés. La solution retenue est un appariement SIFT basé sur les cartes de relief des faces. Les solutions sont ensuite filtrées en deux étapes de manière à conserver les liens utiles au remontage de l'objet.

Les résultats montrent que la méthodologie proposée peut être utilisée dans le cadre de la recherche d'appariements d'objets lithiques avec toute fois des limitations dépendant de la rugosité des faces. Deux remontages ont été effectués dans le cadre des tests. Dans le premier cas, le remontage complet peut être effectué. Dans le deuxième cas, les pièces les plus petites et les plus lisses ne sont pas remontées avec le reste des pièces.

**Mots-clés :** Remontage archéologique, SIFT, appariement, assemblage automatique, obscurcissement volumétrique.



# Abstract

Reassemblies in lithic archaeology are precious tools that allow us to deduce information about archaeological sites that allow us to better understand the civilizations that lived there tens of thousands, even hundreds of thousands of years before us. Today, these reassemblies are mainly done manually. It is a long and tedious work that requires specific personnel. It is within this framework that the present work is inscribed with the aim of developing an automated digital matching method.

The methodology presented in this work proposes different matching techniques that can allow the reassembly. The clouds settings using 3D interest points and 2D images of the faces are compared. The chosen solution is a SIFT matching based on the relief maps of the faces. The solutions are then filtered in two steps in order to keep the links useful for reassembling the object.

The results show that the proposed methodology can be used in the context of lithic object matching, but with limitations depending on the roughness of the faces. Two reassemblies were carried out within the framework of the tests. In the first case, the complete reassembly can be carried out. In the second case, the smallest and smoothest pieces are not reassembled with the rest of the pieces.

**Keywords :** Archaeological reassembly, SIFT, matching, automatic assembly, volumetric obscuration.



# Table des matières

Résumé	I
Abstract	III
Remerciements	VII
Abréviations	IX
Introduction	1
<b>1 État de l'art</b>	<b>3</b>
1.1 Les remontages en archéologie . . . . .	3
1.2 Les appariements en archéologie lithique . . . . .	4
1.3 Fondamentaux de l'appariement 3D . . . . .	5
1.3.1 Calages grossiers . . . . .	5
1.3.2 Calage fin . . . . .	11
1.4 Acquisition de données . . . . .	11
<b>2 Méthodologie</b>	<b>13</b>
2.1 Jeux de données de démonstration . . . . .	14
2.2 Découpage en faces . . . . .	14
2.2.1 Calcul des normales des points . . . . .	15
2.2.2 Découpage des faces . . . . .	15
2.2.3 Comparaison des deux méthodes de découpage de faces . . . . .	17
2.3 Calage grossier . . . . .	18
2.3.1 Appariement à l'aide de SIFT . . . . .	18
2.3.2 Appariement à l'aide d'ISS . . . . .	22
2.4 Filtrage des solutions . . . . .	24
2.4.1 Filtrage des solutions SIFT . . . . .	24
2.4.2 Filtrage des solutions ISS . . . . .	26
2.5 Remontage de l'objet . . . . .	29
<b>3 Résultats</b>	<b>31</b>
3.1 Présentation des jeux de données . . . . .	31
3.1.1 Lokalalei 2C . . . . .	31
3.1.2 MIBR16 . . . . .	32
3.1.3 JGBO1 . . . . .	32
3.2 Test des différents paramètres . . . . .	33
3.2.1 Découpage des faces par propagation des normales . . . . .	33
3.2.2 Création d'images . . . . .	34
3.2.3 Filtrage des solutions . . . . .	37

3.2.4	Densité du nuage de points . . . . .	38
3.3	Déroulement de la chaîne de traitement . . . . .	40
3.4	Limitations . . . . .	44
<b>Conclusion générale</b>		<b>47</b>
<b>A Annexes techniques</b>		<b>49</b>
A.1	Vue 3D de l'extraction de faces par trois points . . . . .	49
A.2	Vue 3D de l'extraction de faces par propagation des normales . . . . .	52
A.3	Test de découpage en faces par propagation des normales . . . . .	55
A.4	Vue 3D du résultat du test de découpage en faces . . . . .	57
A.5	Vue 3D du remontage complet de Lokalalei 2C . . . . .	59
A.6	Vue 3D du remontage partiel de MIBR16 . . . . .	61
<b>B Mode d'emploi</b>		<b>63</b>
B.1	Guide utilisateur . . . . .	63
B.1.1	Installation . . . . .	63
B.1.2	Utilisation . . . . .	64
B.1.3	Conseil . . . . .	64
B.2	Guide programmeur . . . . .	65
B.2.1	Description des fonctions . . . . .	65
B.2.2	Pistes d'améliorations . . . . .	66

# Remerciements

Je tiens particulièrement à remercier, pour le suivi, l'aide, les remarques constructives sur les aspects théoriques et pratiques ainsi que leur disponibilité, les personnes suivantes :

— Monsieur Adrien Gressin, Docteur ENSG, professeur à la HEIG-VD et directeur de ce travail, pour sa disponibilité et le temps consacré tout au long de ce travail de master ;

— Monsieur Xavier Muth, Project Manager à Get in Situ, collaborateur scientifique à la HEIG-VD et mandant de ce travail, pour sa disponibilité et le temps consacré tout au long de ce travail de master ;

— Monsieur Matthieu Deveau, Docteur ENSG, Computer Vision Software Engineer à Pix4D, pour les conseils et remarques lors de la défense intermédiaire ;

— Le Centre Archéologique de Pincevent (CAP), pour le prêt des données du remontage de deux blocs-sièges. Ces deux éléments sont issus de l'habitat n 1 de Pincevent fouillés en 1964.

— Mesdames Catherine Cretin, conservatrice et Nathalie Fourment, directrice du Musée National de Préhistoire (les Eyzies de Tayac, 24), pour le prêt des données du remontages des moulages du remontage Lokalalei 2C.

— Monsieur Jean-Guillaume Bordes, Maître de conférence à l'Université de Bordeaux, pour la mise à dispositions de débitages expérimentaux réalisés dans le cadre de travaux pédagogiques.

— Monsieur Michel Brenet, de l'Institut National de Recherches Archéologiques Préventives (INRAP), pour le prêt des données des débitages expérimentaux MIBR, numérisés par le Max Planck Institute.

ainsi que toutes les autres personnes qui m'ont soutenu de près ou de loin.





# Abréviations

<b>AP</b>	: Avant le présent
<b>ISS</b>	: Intrinsic Shape Signatures
<b>SHOT</b>	: Signature of Histogrammes of Orientations
<b>CSHOT</b>	: Color Signature of Histogrammes of Orientations
<b>SIFT</b>	: Scale-invariant feature transform
<b>SVF</b>	: Sky View Factor
<b>VO</b>	: Obscurcissement Volumétrique
<b>VOP</b>	: Obscurcissement Volumétrique Positif
<b>VON</b>	: Obscurcissement Volumétrique Négatif
<b>ICP</b>	: Iterativ Closest Point
<b>RANSAC</b>	: Random Sample Consensus

# Introduction

## Contexte de l'étude

En archéologie, autant les informations que l'objet d'étude des scientifiques sont (très) souvent morcelés. Lors de fouilles archéologiques, les objets retrouvés sont souvent cassés et des morceaux sont manquants. De même, les informations tirées des objets trouvés par les archéologues ne sont qu'une partie de la réalité. Ici, nous nous intéressons aux objets physiquement fracturés intentionnellement ou non et découverts comme tels. L'objet du travail est d'aider à l'appariement 3D de ces objets fracturés, dans le but de pouvoir les reconstruire. L'intérêt du remontage ne se trouve pas dans l'objet remonté en lui-même, mais dans les informations sur la stratigraphie et les interactions entre sites que le remontage apporte. Dans l'optique de récolter des informations sur le site, il peut être aussi important de prouver l'absence d'appariement que de trouver des remontages.

En Suisse, la grande majorité des sites archéologiques sont postérieurs à la dernière ère glaciaire soit environ 10'000 ans AP, ce qui correspond à la fin de la période paléolithique ainsi qu'aux périodes mésolithique et néolithique. Les glaciers ont rasé tout ce qui est antérieur à l'exception de quelques sites dans des grottes qui sont plus anciens. Pour comprendre la vie, les interactions et l'organisation sur ces sites, les archéologues cherchent à déduire des informations des objets trouvés sur place. Dans le cadre de ces travaux, on cherche à faire des remontages d'objets pour connaître la forme des outils utilisés, mais aussi les interactions au sein du site, voir avec d'autres sites. Par exemple, dans le cas d'objets en pierres taillées, le remontage d'éclats de débitage avec l'objet final peut communiquer des informations précieuses sur le lieu de débitage et celui d'utilisation de l'objet et des interactions qu'il peut y avoir entre les sites. Pour les archéologues, il est primordial de pouvoir trouver des appariements entre objets pour comprendre au mieux le site.

## Problématique

Le remontage des objets en pierres taillées dans le cadre de fouilles archéologiques constitue une source d'informations importante pour les archéologues. Malheureusement, il s'agit de travaux longs et compliqués car ils sont encore effectués à la main. D'une part, les moyens humains et financiers ne peuvent pas toujours être engagés pour effectuer ces travaux, et d'autre part, il est très compliqué de le faire entre des fouilles archéologiques géographiquement proches mais qui sont décalées dans le temps.

Dans notre cas, l'étude se portera sur le remontage de manière numérique d'objets en pierres taillées. Dans ce cas, toutes les faces sont potentiellement appariables à d'autres. Le but est d'arriver à une méthode de remontage 3D basée sur la géométrie des objets en partant du principe que deux faces communes ont une géométrie semblable qui peut s'apparier en utilisant des méthodes de recalage 3D existantes. Pour y parvenir, il sera nécessaire d'avoir une reconstruction 3D numérique des différentes pièces à apparier. Les aspects d'acquisition des données et de reconstructions numériques de celles-ci ne sont pas directement traités dans ce travail. Néanmoins, la question de la précision et la résolution d'acquisition sera abordée car l'appariement en dépend. Le volet principal de l'étude traitera de l'appariement d'objets ayant des faces communes. La méthode utilisée devra permettre d'assembler les objets qui peuvent l'être et d'exclure les assemblages entre objets qui n'ont pas de faces communes. Pour arriver à ces assemblages, ce travail explorera les voies d'assemblage entre nuages de points et par transformation des nuages de points en carte de profondeur dans le but d'assembler ces dernières.

# Chapitre 1

## État de l'art

Ce chapitre présente différentes solutions de remontage existantes et les méthodes utilisées pour chacune. Sur cette base, il présente une palette d'outils utilisables pour l'appariement d'objets et les données sur lesquelles ils sont applicables. Enfin, il passe en revue les acquisitions possibles des données nécessaires à ces appariements .

### 1.1 Les remontages en archéologie

En archéologie, différents types de remontages sont possibles en fonction des objets concernés. Différents types d'objets, tels que poteries, ossements, pierres, etc., peuvent être remontés dans le but de recueillir des informations sur les civilisations auxquelles ils ont appartenu. Dans [Kampel and Sablatnig, 2004], les auteurs traitent du remontage d'objets en céramique par une approche 3D du problème. Après avoir déterminé les appariements possibles en prenant en compte des critères de diamètre et de hauteur de l'objet à partir des fragments, ils appariant les fragments en superposant les axes de rotations des deux fragments et en calculant les écarts minimaux points à points entre les deux plans fractures avec 2 degrés de liberté (rotation autour de l'axe et translation le long de l'axe) (Figure 1.1a). Toujours en remontage de céramiques, les auteurs de [Üçoluk and Hakkı Toroslu, 1999] utilisent des vecteurs de courbure et de torsion des plans de cassure pour en faire une chaîne de caractère et appliquer un algorithme d'appariement sur ces chaînes qui s'apparente à une résolution 2D du problème (Figure 1.1b). Bien que des méthodes numériques soient possibles, les appariements manuels de poteries peuvent encore être utilisés lorsque les pièces ne sont pas numérisées (Figure 1.1c).

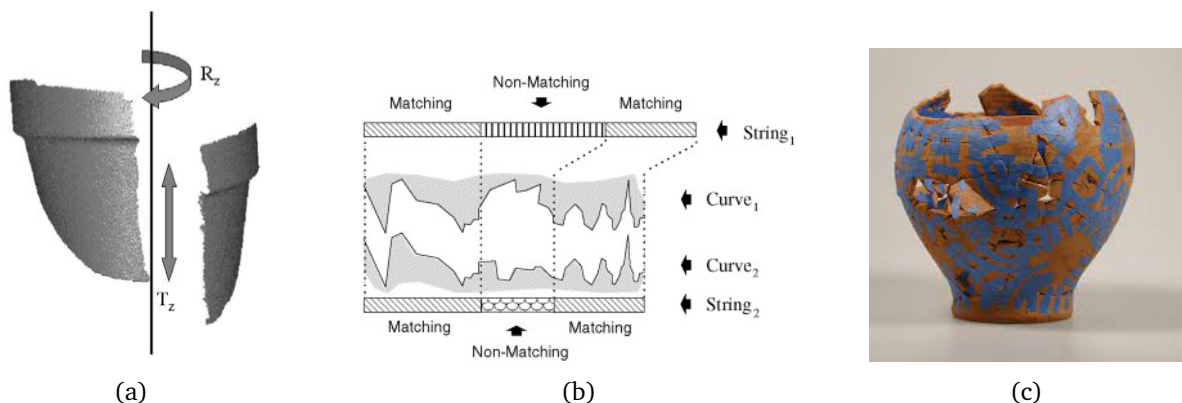


FIGURE 1.1 – Différentes méthodes de remontage de poterie.

A l'intersection entre le domaine de l'imagerie médicale et du remontage d'objet, des travaux semblables ont également été menés. Les auteurs de [Liao et al., 2020] propose une solution de réduction de fracture osseuse (Figure 1.2a) pouvant également faire de l'appariement multi-objets (Figure 1.2b). Mais les tests effectués montrent tout de même que l'assemblage de pièces multiples est possible mais qu'il est fortement ralenti par les choix d'assemblage qui ont été faits.

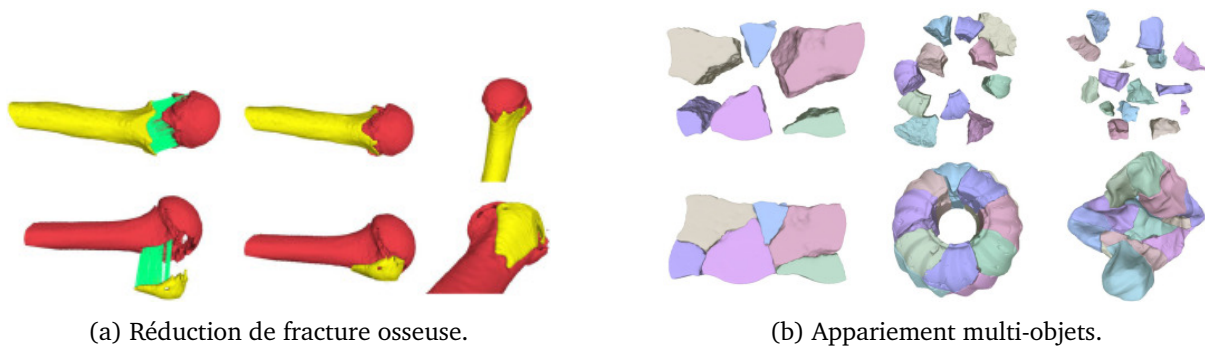


FIGURE 1.2 – Appariement en imagerie médicale.

## 1.2 Les appariements en archéologie lithique

Actuellement, le remontage d'objets en pierre est encore fait de manière manuelle, ce qui le rend long et fastidieux. Parfois, le remontage n'est tout simplement pas fait par manque de temps ou de financement. Lors de fouilles archéologiques, la texture et la couleur des objets retrouvés sur les sites sont rarement identifiables car elles sont altérées par le séjour en terre (patine). En revanche, la géométrie est rarement altérée. Il est donc plus facile de travailler sur ce volet. Dans l'approche du remontage géométrique d'objets, la différence réside dans le fait que les morceaux ont pour certaines catégories des faces externes dont on sait qu'elles ne sont pas à assembler et d'autres dont on ne sait rien. Par exemple, la reconstruction de poterie peut facilement être assimilée à un remontage 2D qui se fait sur les plans de cassure. De même, la connaissance pour certains objets de la forme finale de l'objet aide aussi grandement au remontage. Dans le cas de remontages lithiques, les surfaces de contact sont les faces de chaque pierre. Aucun pré-calage entre les objets n'est connu avant appariement et il n'y a pas de possibilité de fixer l'un ou l'autre des paramètres de manière à réduire le problème en 2D. Il sera donc nécessaire d'utiliser une méthode d'appariement 3D.



FIGURE 1.3 – Image d'illustration : Remontage lithique. Source : [www.paleotime.fr](http://www.paleotime.fr)

## 1.3 Fondamentaux de l'appariement 3D

### 1.3.1 Calages grossiers

L'appariement 3D se déroule généralement en deux étapes. La première consiste en un calage grossier pour laquelle il n'y a pas d'a priori de position relative entre les objets. Cette première étape a pour objectif de déterminer une transformation approximative entre les objets qui pourra ensuite servir de base à un calage fin. Ces calages peuvent être effectués de manière locale, c'est-à-dire par paires, ou de manière globale avec une répartition des erreurs sur l'ensemble des objets à caler. Un calage global est avantageux lorsqu'il y a de nombreux recouvrements entre les nuages afin d'obtenir une solution qui prenne en compte l'ensemble des liens. Un calage local sera à l'inverse plus utile lorsque les objets à appairer vont principalement par paires. A noter qu'un remontage global peut également être fait en assemblant les différents calages locaux par des successions de transformations entre nuages. Plusieurs méthodes peuvent être utilisées pour effectuer des calages grossiers et les données en entrée peuvent également varier. Ci-dessous, quelques méthodes de calage entre nuages de points et entre images qui pourraient être utilisées dans le cadre de ce projet sont présentées.

#### Méthodes de calage 3D

**Spin-images :** La méthode des spin-images [Johnson, 1997] (Figure 1.4) est caractérisée par des points d'intérêt "orientés". Les points ont des normales calculées par rapport à la surface locale. Ces normales sont utilisées comme axe de rotation pour la création des spin-images. Les spin-image sont une représentation de surface 3D qui combine la nature descriptive des propriétés des objets globaux avec la robustesse des vues partielles et l'encombrement des descriptions de formes locales. Plus précisément, une base locale est calculée à un point orienté (point 3D avec surface normale) sur la surface d'un objet représenté comme un maillage de surface polygonale. Les positions par rapport à la base d'autres points sur la surface de l'objet peuvent alors être décrites par deux paramètres (coordonnées locales). En accumulant ces paramètres dans un tableau 2D, une spin-image descriptive associée au point orienté est créée. Comme la spin-image code les coordonnées des points sur la surface d'un objet par rapport à la base locale, la description de la forme de l'objet est locale et invariante aux transformations rigides.



FIGURE 1.4 – Huit cadres d'animation qui motivent le nom spin-image. Le processus de génération de spin-image peut être visualisé comme une feuille tournant autour de la normale du point orientée, accumulant des points comme s'il balaye l'espace [Johnson, 1997].

**ISS** : L'algorithme de l'ISS [Zhong, 2009] (Figure 1.5) est un descripteur de forme 3D appelé "Signature de forme intrinsèque". Il caractérise une région locale/semi-locale d'un nuage de points 3D. Deux signatures de formes intrinsèques sont associées en comparant les vecteurs indépendants de forme de vue. Il a l'avantage de fonctionner directement sur les nuages de points 3D et ne nécessite ni maillage en surface ni triangulation des points du nuage. Ces étapes de pré-traitement sont sujettes à erreur pour les données dans les applications réelles, ce qui est typique du bruit du capteur, de la discontinuité de surface due à l'obscurcissement et à l'occlusion, et de la proximité d'objets. Il fait également peu d'hypothèses sur la physique des capteurs ou la géométrie de visualisation.

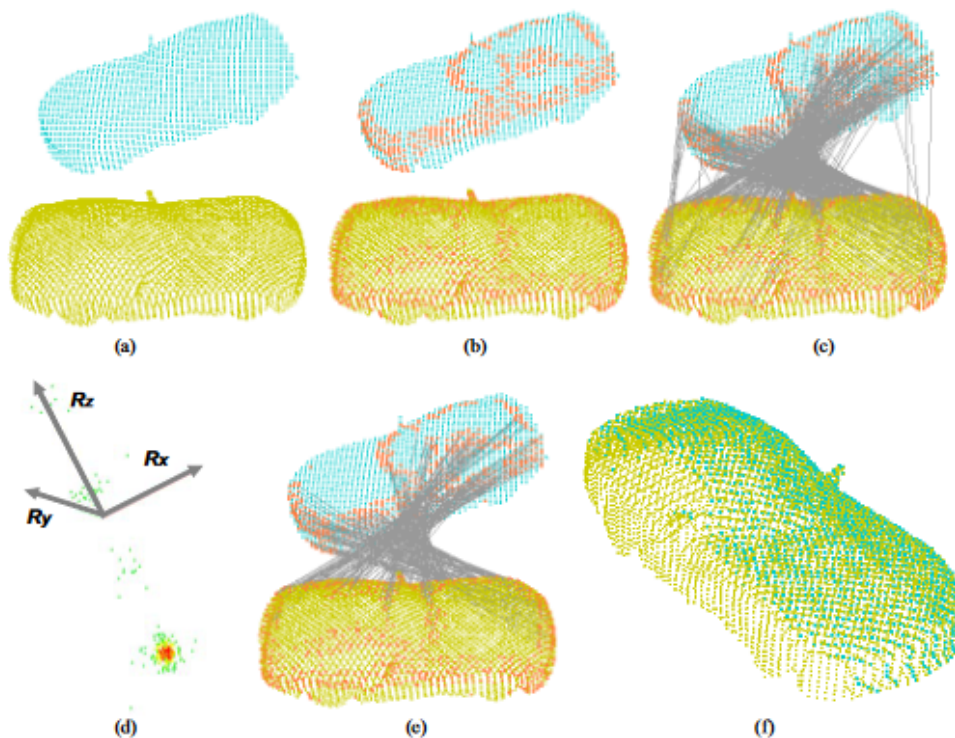


FIGURE 1.5 – Appariement de deux nuages de points à l'aide de l'ISS. (a) nuage de points source de la vue partielle d'une voiture en bleu et le nuage de point de modèle en jaune ; (b) points de base saillants extraits marqués en rouge ; (c) appariement de paires de descripteurs entre la requête et le modèle ; (d) estimation de rotation à partir de paires de descripteurs correspondants : chaque point correspond à la rotation « observation » d'une paire d'appariements de descripteurs de forme intrinsèques, avec la couleur indiquant la densité de l'estimation (vert pour la faible densité et rouge pour la densité élevée) ; (e) appariement de paires de descripteurs correspondant au plus grand groupe de rotations ; (f) alignement du nuage source sur le modèle en utilisant la rotation estimée [Zhong, 2009].

**SHOT** : L'algorithme SHOT [Salti et al., 2014] code les histogrammes des normales dans le descripteur. Le pouvoir discriminant du descripteur est renforcé par l'introduction d'informations géométriques concernant l'emplacement des points dans le descripteur, imitant ainsi une signature. Ceci est fait en calculant un ensemble d'histogrammes locaux sur les volumes 3D définis par une grille 3D superposée sur le support. La grille est alignée avec les axes définis par le cadre de référence local. Ainsi, le descripteur se situe à l'intersection entre Histogrammes et Signatures, d'où son nom Signature of Histogrammes of Orientations (SHOT). Ce descripteur est efficace, descriptif, robuste au bruit. Il permet d'utiliser des nuages de points avec des densités différentes et traite également les couleurs (CSHOT).



**Signatures et Histogrammes :** Les descripteurs sont répartis en deux grandes catégories (Figure 1.6). Les premiers utilisent les signatures, c'est-à-dire qu'ils décrivent la surface locale 3D du voisinage en introduisant un référentiel local et en y encodant des mesures géométriques pour chaque point du voisinage. Les seconds utilisent les histogrammes, c'est-à-dire qu'ils encodent des compteurs d'entités topologiques locales. Les signatures sont potentiellement hautement descriptives grâce à l'utilisation d'informations spatialement bien localisées, mais de petites erreurs dans la définition du cadre de référence local ou de petites perturbations dans le caractère codé peuvent modifier considérablement le descripteur final. En termes généraux, si on les compare aux signatures, les histogrammes compensent le pouvoir descriptif par la robustesse [Salti et al., 2014].

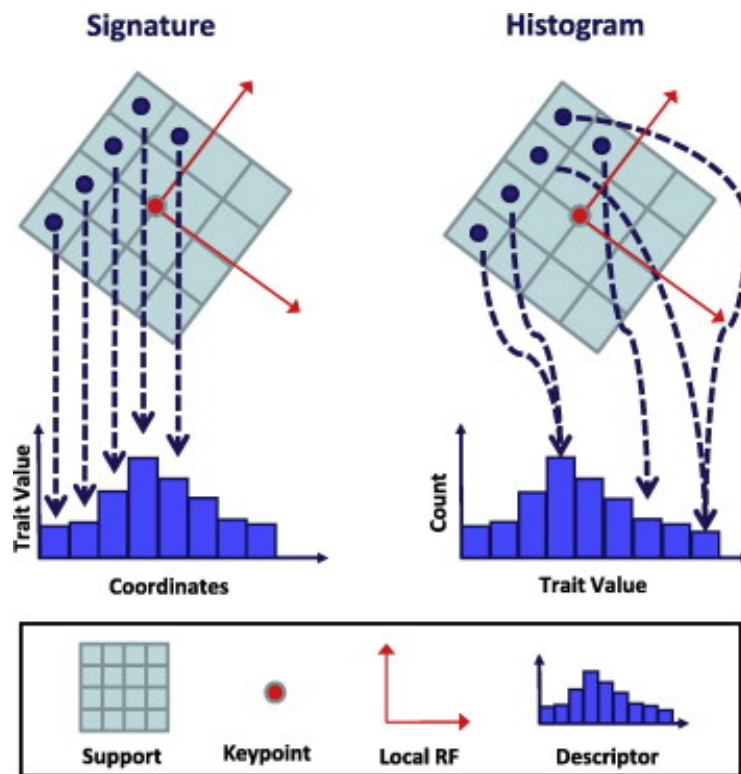


FIGURE 1.6 – Schéma de construction des descripteurs des catégories de signatures et d'histogrammes [Salti et al., 2014].

## Méthodes de calage 2D

**SIFT :** La méthode SIFT extrait des caractéristiques invariantes distinctives des images qui peuvent être utilisées pour effectuer une correspondance fiable entre différentes vues d'un objet ou d'une scène. Les caractéristiques sont invariantes pour l'échelle et la rotation de l'image, et démontrent une correspondance robuste à travers une gamme substantielle de distorsion affine, changement de point de vue 3D, ajout de bruit et changement d'éclairage. Les caractéristiques sont très distinctives, en ce sens qu'une seule caractéristique peut être correctement associée à une probabilité élevée par rapport à une grande base de données de caractéristiques de nombreuses images. La reconnaissance se fait en faisant correspondre des caractéristiques individuelles à une base de données de caractéristiques d'objets connus à l'aide d'un algorithme du plus proche voisin, suivi d'une transformation de Hough pour identifier les grappes appartenant à un seul objet et enfin en effectuant une vérification par moindres carrés pour des paramètres cohérents. Cette approche de la reconnaissance permet d'identifier avec robustesse les objets en obtenant un rendement presque en temps réel [Lowe, 2004].

### Utilisation de données 3D avec méthode de calage 2D

**Création de cartes de profondeur :** Dès lors que l'utilisation d'un algorithme de calage 2D pour la mise en correspondance de données 3D est envisagée, la question de la transformation des données dans un format utilisable pour l'algorithme se pose également. Les données 3D peuvent être représentées sur un support 2D tout en conservant une information altimétrique (2D+1D), c'est le cas par exemple avec les courbes de niveaux ou les modèles numériques de terrain (MNT). A partir d'un nuage de points, une grille régulière peut être créée pour chaque face avec pour chaque pixel une valeur correspondant à la donnée altimétrique du nuage de point. En d'autres termes, la création d'une carte de profondeur (Figure 1.7) perpendiculaire à un plan moyen de la face pour chaque face de l'objet.

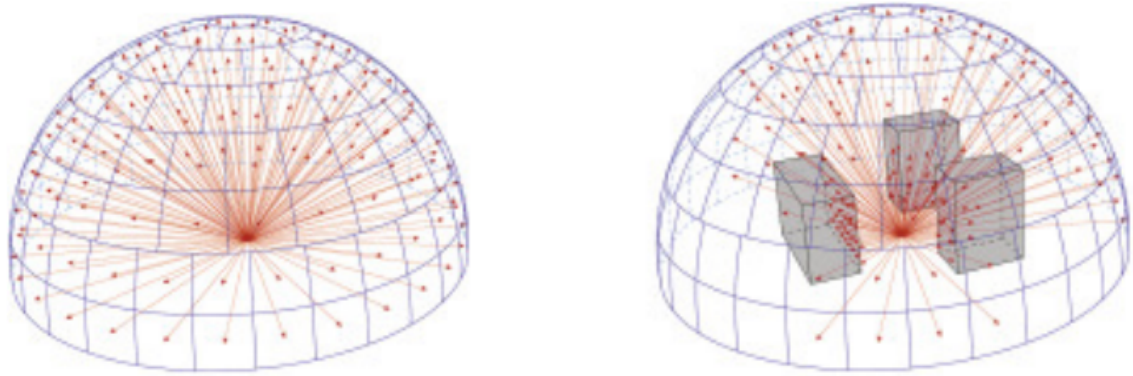


FIGURE 1.7 – Carte de profondeur d'une face de fragment rocheux

Néanmoins, pour que les différentes images créées puissent s'apparier, elles doivent avoir en commun des points d'intérêts qui soient identiques dans les deux images qui doivent s'apparier. Or, dans le cas d'une carte de profondeur, les valeurs d'altitudes qui remplissent la grille régulière ne peuvent pas être considérées comme points potentiellement appariables car ils sont dépendants du plan dans lequel la carte est créée et de la valeur absolue d'altitude des points du nuage. Pour pallier à ce problème, il faudrait utiliser le relief local plutôt que les valeurs absolues d'altitude. Plusieurs outils existent pour représenter le relief. On peut citer les ombrages, le Sky View Factor [Yi and Kim, 2017], ou l'Obscurcissement Volumétrique (VO) [Rolland et al., 2022].

**Ombrages :** Dans le cas d'un appariement géométrique entre deux images dont on ne connaît pas l'orientation relative, il est important que l'ombrage utilisé ne soit pas dépendant d'une direction. Pour pallier à ce problème, un ombrage multi-directionnel peut être utilisé. De même, pour que l'ombrage soit identique sur les deux faces à appairer, il faut que les plans de référence soient colinéaires une fois les pièces assemblées, mais l'orientation relative des pièces et donc des plans avant calage n'est pas connue. La méthode est limitée par rapport à cette dépendance à l'orientation du plan de référence, mais peut fonctionner dans le cas particulier où les plans sont approximativement colinéaires après calage.

**Sky View Factor :** Le Sky view factor (SVF) détermine la portion de ciel visible à partir de chaque point de l'image à l'intérieur d'une demi-sphère au-dessus de l'horizon (Figure 1.8). Pour un point bas du modèle, le SVF calculé ne dépend pas du plan de référence car les rayons sont bloqués par le terrain avoisinant. En revanche, le plan de référence choisi peut avoir une influence lorsque l'on calcule le SVF d'un point haut du modèle où les rayons ne seront bloqués par le terrain que dans certaines orientations du plan de référence.



(a) Nombre total de rayons (145 rayons).

(b) Nombre de rayons non obstrués (87 rayons).

FIGURE 1.8 – Détermination du Sky view factor pour un point donné [Yi and Kim, 2017].

**Obscurcissement Volumétrique :** Le VO calcule le volume d'une sphère au-dessus de la surface du terrain par rapport au volume total de la sphère (Figure 1.9b). Deux variantes de calcul sont également proposées avec l'obscurcissement volumétrique positif (VOP) (Figure 1.9c) et l'obscurcissement volumétrique négatif (VON) (Figure 1.9d) qui ne prennent en compte qu'un seul hémisphère. Les calculs des VOP et VON basés sur une demi-sphère sont dépendants du plan de référence, alors que celui du VO, basé sur une sphère complète s'en affranchi complètement.

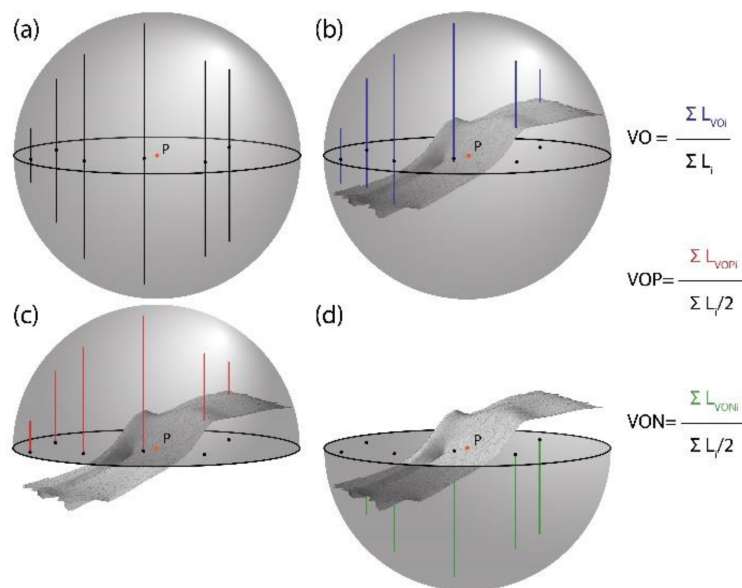


FIGURE 1.9 – Détermination du VO, VOP, VON pour un point donné [Rolland et al., 2022].

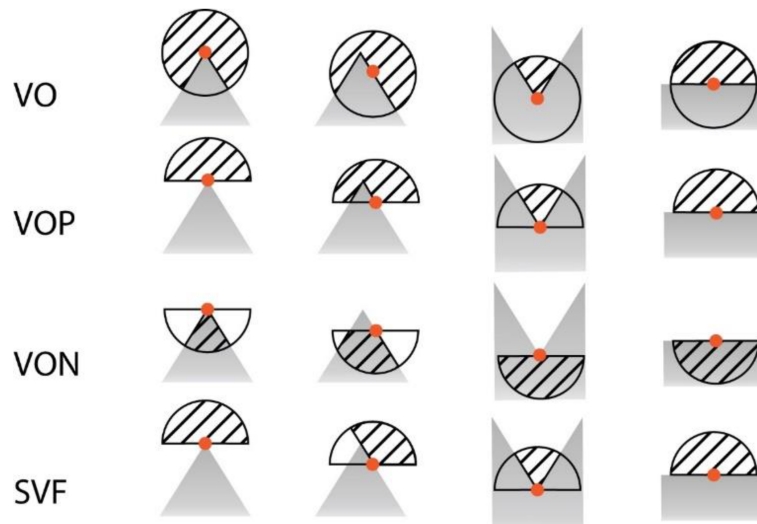


FIGURE 1.10 – Comparaison entre VO, VOP, VON et SVF pour une application dans différents cas donnés [Rolland et al., 2022].

En résumé, dans les méthodes proposées ci-dessus (Figure 1.10), seul le VO propose un traitement rigoureux permettant d'obtenir un résultat identique depuis deux faces qui s'assemblent (hors imprécisions de mesures). Les autres méthodes sont utilisables dans des cas particuliers ou avec un nombre de points potentiellement appariables réduit. Bien que ça ne soit pas un critère éliminatoire, le fait que le script du VO soit en Python est également un plus pour le développement du projet qui est également prévu en Python (Table 1.1).

Méthode	Dépendant du plan de référence	Implémentation
Ombrage	Oui	QGis, GDAL, etc...
SVF	En partie	SAGA
VOP	En partie	Python
VON	En partie	Python
VO	Non	Python

TABLE 1.1 – Le tableau récapitule les dépendances ou non au plan de référence choisi pour la face à transformer et les outils sur lesquels sont implémentés la méthode.

### Vérification de la cohérence géographique des points d'appariements

Lors de la mise en correspondance de paires de points à l'aide d'une méthode d'appariement utilisant des points d'intérêts en 2D tel que SIFT, il ressort des paires de points homologues entre deux objets qui peuvent potentiellement s'apparier. Afin de déterminer quelles paires de points détectées sont réellement utilisables pour l'appariement et si les deux objets sont appariables, il est nécessaire de vérifier les appariements potentiels d'après un modèle mathématique. Une vérification est possible sur l'homographie entre les deux images ou sur la transformation 3D entre les deux nuages. Une solution pour cette vérification consiste à utiliser l'algorithme RANSAC [Fischler and Bolles, 1981] pour déterminer le meilleur ensemble de points parmi les correspondances de points d'intérêt trouvées.

**RANSAC (Random Sample Consensus) :** L'algorithme RANSAC est basé sur un choix au hasard d'un minimum de points permettant de résoudre les paramètres du modèle. Les résultats sont ensuite comparés à une tolérance prédéfinie. Si la tolérance n'est pas atteinte, l'expérience est réitérée au maximum le nombre de fois prédéfini par les paramètres en entrée.

### 1.3.2 Calage fin

Une fois que l'orientation relative entre les deux nuages est calculée à l'aide de données réduites du nuage de points, un calage fin des nuages peut être effectué. L'algorithme le plus utilisé pour cette opération est l'ICP [Besl and McKay, 1992].

**ICP (Iterativ Closest Point) :** Contrairement aux algorithmes de calage grossier, l'ICP nécessite une solution approchée du résultat en entrée car il va simplement converger vers un minima local en optimisant les rotations et translations données. Ces optimisations sont faites en minimisant la distance moyenne au carré entre points. Le calcul de ces distances peut se faire de deux façons : la première, point-à-point minimise directement la distance entre points alors que la seconde, point-à-plan minimise la distance entre les points du premier nuage et le plan du second le long de la normale du point. L'utilisation de l'une ou l'autre méthode sera plus adaptée en fonction de la forme globale du nuage et de l'utilisation faite. Cet algorithme est implémenté sur la librairie Open3D [Zhou et al., 2018]. Il est proposé avec un paramètre supplémentaire permettant de donner une distance maximum de correspondance des points, ce qui permet de caler également des nuages dont seule une petite partie de chacun des nuages est en contact.

## 1.4 Acquisition de données

Pour pouvoir utiliser les méthodes décrites ci-dessus, il est nécessaire de procéder à une acquisition 3D des données. Comme la couleur et la texture sont rarement utilisables sur les objets retrouvés, c'est la géométrie du modèle numérique qui sera importante. Les acquisitions sont le plus souvent faites à l'aide de scanners laser. La photogrammétrie peut également être utilisées pour les acquisitions.

**Acquisition par scanner laser 3D :** Dans le cadre de fouilles archéologiques, des relevés 3D à l'aide de scanner laser peuvent être effectués à titre de documentations. Plusieurs types de scanner peuvent être utilisés. Les critères de choix principaux sont la taille de l'objet à scanner, la précision désirée et le temps à disposition. Des scanner différents seront utilisés s'il s'agit d'ouvrages bâtis, d'ossement ou de petits objets. Le coût d'acquisition des scanners et logiciels de traitement sont relativement élevés, mais le temps de traitement pour obtenir des données sera faible pour la plupart des scanners. La précision atteignable des scans pour les objets lithiques est de quelques centièmes de millimètres.

**Acquisition photogrammétrique :** L'acquisition de données photogrammétriques dans le cadre de fouilles archéologiques est de plus en plus courante et offre aux archéologues des outils supplémentaires de travail. Le développement des techniques d'acquisition photogrammétriques offre de nouvelles possibilités en termes de remontage numérique et même comme cité dans "Ceramics Fragments Digitization by Photogrammetry, Reconstructions and Applications" [Barreau et al., 2014] d'impression 3D des objets. L'accessibilité et la facilité d'utilisation des capteurs font de la photogrammétrie un outil de choix pour les travaux liés à l'archéologie. Néanmoins, pour obtenir un résultat de qualité, il est nécessaire que l'acquisition suive quelques règles élémentaires concernant la géométrie et la prise des photos. L'article "3 x 3 rules for simple photogrammetric documentation of architecture"[Waldhäusl and Ogleby, 1994] recense les points importants devant être respectés, notamment concernant la géométrie d'acquisition, certains points en revanche ne concernent plus les clichés numériques. Contrairement au scanner laser, le temps de traitement pour obtenir des données 3D depuis une acquisition photogrammétrique peut être conséquent. Il nécessite également quelques connaissances de base en photogrammétrie.



## Chapitre 2

# Méthodologie

Ce chapitre présente les différentes étapes de traitements nécessaires pour passer des modèles numériques 3D d'objets en pierre sans a priori de position à un remontage complet de la pièce. Les différentes sections du chapitre se rapportent aux principales étapes du processus de remontage : découpage en faces, calage grossier, filtrage des solutions et remontage de l'objet. Pour des raisons d'optimisation du temps à disposition, le choix a été fait de ne tester, parmi les méthodes recensées dans la section 1.3, que les méthodes implémentées sur une librairie python ou dont le code python est disponible. La méthodologie traitera deux chaînes de traitements. La première utilise un calage grossier de nuages à l'aide de l'ISS qui est disponibles sur la librairie Open3D. La deuxième utilise un appariement de faces via SIFT qui est disponible sur OpenCV.



FIGURE 2.1 – La chaîne de traitement utilisée pour le remontage des objets.

## 2.1 Jeux de données de démonstration

Pour illustrer la méthodologie avec un exemple simple, un jeu de données de deux blocs-sièges<sup>1</sup> (Figure 2.2) sera utilisé. Il s'agit de deux blocs d'environ 20cm X 10cm X 6cm avec chacun environ 1'500'000 points. Il a été transmis sous forme d'un maillage avec une résolution d'environ 0,3mm.

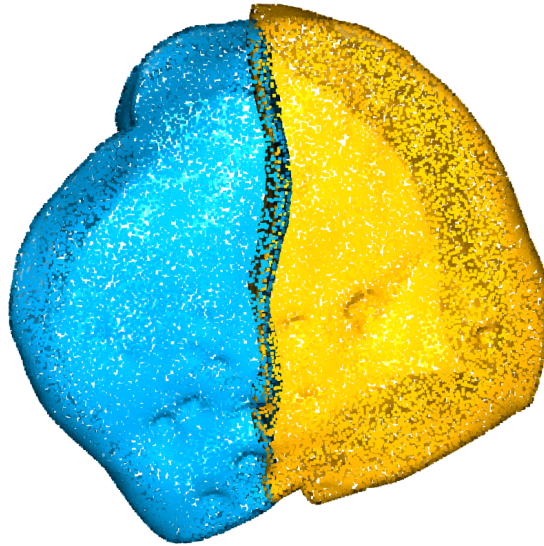


FIGURE 2.2 – Jeu de données de deux blocs-sièges utilisé pour illustrer la méthodologie.

## 2.2 Découpage en faces

Dans l'optique de découper automatiquement les nuages en faces, deux méthodes sont proposées afin de pouvoir sortir un découpage adapté à l'appariement qui suivra. La première se base sur la détection du meilleur plan par trois points dans le nuage et la seconde sur la propagation de normales. Ces méthodes nécessitant que chaque point ait une normale tangente au plan local orientée vers l'extérieur de la pièce, le calcul de ces normales est également traité ci-dessous.

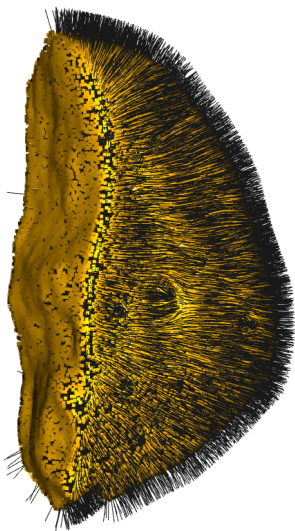
---

1. Remontage de deux blocs-sièges appartenant au Centre Archéologique de Pincevent (CAP). Ces deux éléments sont issus de l'habitat n 1 de Pincevent fouillés en 1964.

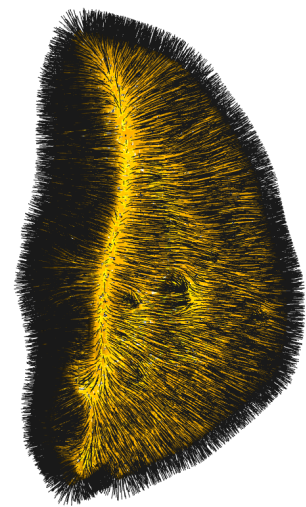


### 2.2.1 Calcul des normales des points

La librairie Open3D gère les normales et des outils adéquats pour leur création (Figure 2.3a) et orientation (Figure 2.3b) y sont disponibles. L'ajout des normales se fait en deux étapes. Premièrement, les normales sont créées pour chaque point, puis toutes les normales sont orientées vers l'extérieur de la pièce. Cette deuxième opération prend en compte un nombre de points voisins pour le calcul du plan tangent qui peut être défini par l'utilisateur. Plus le nombre de points utilisé est petit, plus le plan tangent est défini localement. Un nombre de point trop faible peut entraîner une modification très rapide de l'orientation du plan tangent en cas de point faux dans le proche voisinage, mais un nombre de point trop important peut entraîner le calcul d'un plan avec des points ne faisant pas partie du voisinage. Les pièces fines sont les principales sources d'erreurs possibles pour cette étape. Lorsqu'une orientation de normale doit être faite sur une pièce fine, il faut veiller à ce que les points du voisinage qui doivent être pris en compte pour le calcul du plan tangent ne soient jamais plus éloignés que les points de la face opposée.



(a) Nuage et normales après création.



(b) Nuage et normales après orientation des normales vers l'extérieur de la pièce.

FIGURE 2.3 – Création et orientation des normales pour un nuage de points.

### 2.2.2 Découpage des faces

Le but du découpage est d'obtenir des faces de chaque bloc qui soient appariables avec celles découpées sur d'autres blocs. Les faces doivent se couper aux arêtes franches, ne pas présenter de surplomb et avoir le moins de trou possible à l'intérieur. Deux méthodes sont proposées ci-dessous pour découper les blocs en faces.

### Extraction de plans par trois points

Une fonction de détection du meilleur plan est disponible dans Open3D. Elle utilise l'algorithme RANSAC pour itérer un nombre choisi de fois des plans par trois points. A partir de ces plans, une distance maximum d'inclusion des points dans le plan est définie. Le plan retenu est celui qui inclus le plus grand nombre de points. Une fois cette première étape terminée, un angle limite par rapport à la normale du plan est défini. La normale de chaque point faisant partie du plan est testée par rapport à cet angle limite et seuls les points dont la normale à un angle inférieur à l'angle limite avec la normale du plan sont conservés dans le plan (Figure 2.4). Les points du plan sont soustraits au nuage et l'opération est répétée jusqu'à ce que tous les plans ayant un nombre de points minimum défini par l'utilisateur soient extraits. Un exemple 3D d'extraction de plans par 3 points est disponible en Annexe A.1.

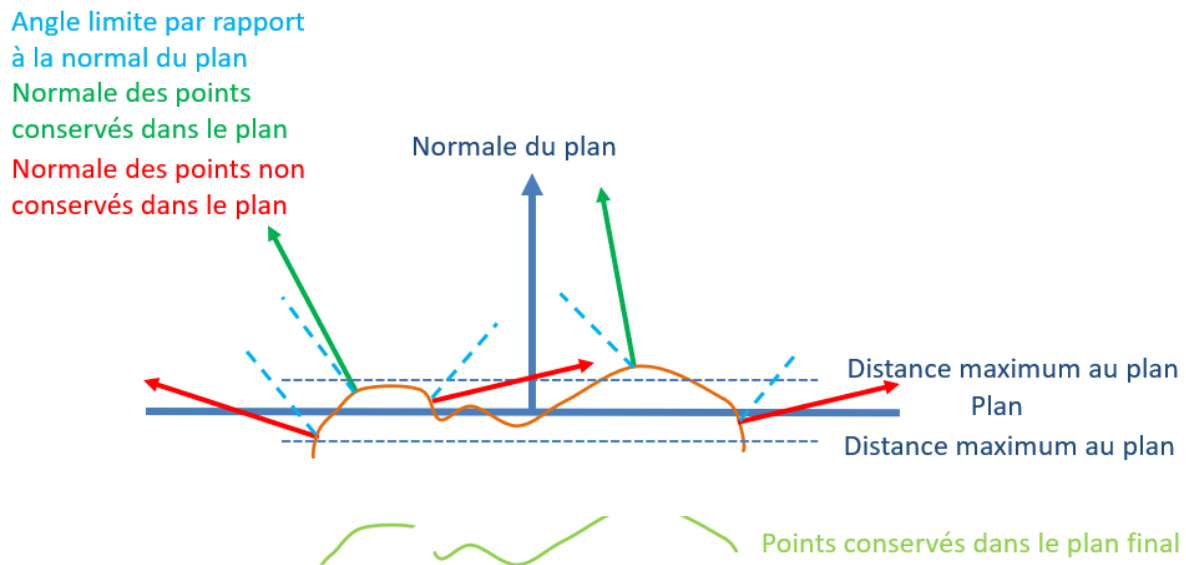


FIGURE 2.4 – Coupe de principe d'extraction des plans avec la méthode de détection du meilleur plan par 3 points.

### Extraction de faces par propagation des normales

La méthode d'extraction de face par propagation des normales part d'un point quelconque du nuage pour y ajouter de proche en proche les points répondants aux critères de la face. Le premier critère est la distance par rapport à un point considéré comme faisant partie de la face. Le deuxième critère est l'angle maximal par rapport à la moyenne des normales déjà retenues dans la face. Un critère de nombre de points minimum est également fixé afin de valider le groupe de points comme étant une face appariable. Ces trois paramètres seront testés au chapitre 3. Un exemple 3D d'extraction de faces par propagation des normales est disponible en Annexe A.2.

### 2.2.3 Comparaison des deux méthodes de découpage de faces

Deux gros avantages sont à mettre au crédit de la méthode d'extraction par propagation des normales par rapport à celle de l'extraction de plans par trois points présentée ci-dessus. D'une part, elle s'affranchit du critère de distance maximum au plan qui limite fortement l'extraction de faces incurvées et d'autre part, elle garantit une continuité géométrique de la face. Néanmoins, ces gains qualitatifs se payent en temps de calcul puisque la deuxième méthode prend approximativement huit fois plus de temps que la première pour l'extraction des faces d'une pièce. L'extraction des faces des deux blocs des Annexes A.1 et Annexe A.2, soit deux blocs avec environ 1'500'000 points chacun a pris 89 secondes pour l'extraction de plans par trois points contre 734 secondes pour la propagation de normales<sup>2</sup>. Pour un calcul sur un nombre important de pièces, la phase de découpage des faces peut être parallélisée de manière à réduire le temps de calcul. Sur la face principale de chaque bloc qui doit s'apparier (Figure 2.5a et 2.5b), la différence n'est pas flagrante car elle sont planes et comptent un nombre important de points par rapport au nuage. La différence est beaucoup plus visible sur l'extraction de l'ensemble des faces visible aux Annexes A.1 et A.2. La solution d'extraction de faces par propagation des normales sera conservée pour la suite du travail.



FIGURE 2.5 – Comparaison de l'extraction des faces selon les deux méthodes présentées.

2. Calcul sur une machine avec processeur Intel Core i7 / 16Go de RAM.

## 2.3 Calage grossier

### 2.3.1 Appariement à l'aide de SIFT

Cette méthode doit permettre l'appariement d'images représentant le relief des faces des différentes pièces. Comme mentionné dans la section 1.3.1, les images utilisées pour l'appariement seront le résultat du VO sur la carte de profondeur des faces de chaque pièce. Pour arriver à cette étape, il sera nécessaire de créer des cartes de profondeur à partir des faces.

#### Création des images

**Orientation des faces :** Dans l'optique de la transformation en carte de profondeur des faces, chaque plan subit une rotation permettant d'amener la normale moyenne de la face ( $N$ ), récupérée lors de la création des faces, sur l'axe vertical du système ( $Z$ ).  $N$  est normalisée pour obtenir son vecteur unitaire. Un axe de rotation perpendiculaire à  $Z$  et  $N$  est calculé, ainsi que l'angle de rotation autour de cet axe de  $N$  vers  $Z$ . Cette rotation est ensuite appliquée au nuage de point de la face pour orienter sa normale moyenne à la verticale.

**Définition de la taille maximale des faces en plan :** La dimension maximale de la plus grande face en plan est utilisée comme référence pour la définition de la taille du pixel. Les coordonnées minimum et maximum en  $X$  et en  $Y$  de chaque face sont testées pour déterminer cette dimension maximum. Cette opération permet de conserver le facteur d'échelle entre toutes les faces et de s'assurer que l'ensemble des faces soient représentées entièrement sur l'image.

**Création de la carte de profondeur :** Pour chaque face, une grille régulière en  $X$  et en  $Y$  est créée en fonction de taille maximale des faces déterminées précédemment et du nombre de pixels par côté choisi par l'utilisateur. Cette définition du nombre de pixels par côté fera l'objet de tests au chapitre 3. Cette grille est ensuite remplie avec les coordonnées  $Z$  interpolées linéairement depuis le nuage de point de la face pour créer la carte de profondeur (Figure 2.6).



FIGURE 2.6 – Carte de profondeur d'une face appariable de la pièce test.

**Création du VO :** Comme on peut le voir sur la Figure 2.6, la carte de profondeur est peu contrastée et les reliefs locaux de la face ne sont pas visibles. Pour faire ressortir ces reliefs, un VO (Figure 2.7) est calculé comme expliqué dans la section 1.3.1.

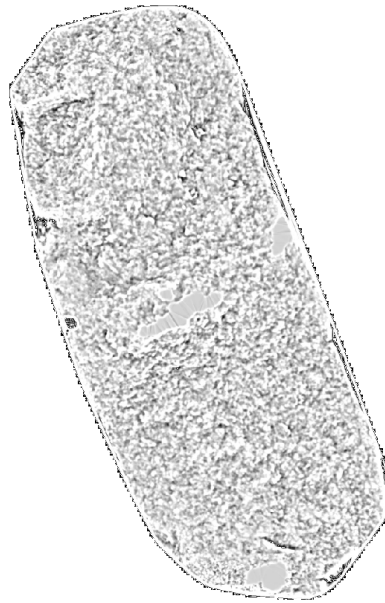


FIGURE 2.7 – VO calculé sur la base de la carte de profondeur de la Figure 2.6.

Pour les besoins de l'appariement, il faut que le même objet soit représenté. Dans le cas des appariement de faces, cela signifie voir la même face depuis l'extérieur d'une pièce et l'intérieur de l'autre comme illustré dans la Figure 2.8 ci-dessous.

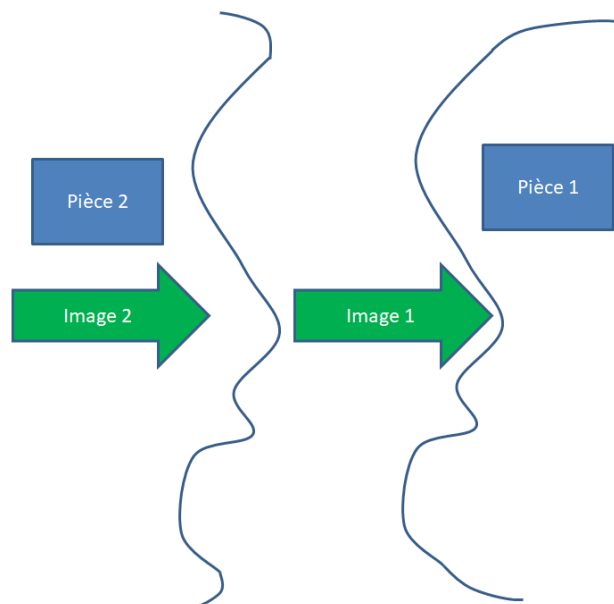


FIGURE 2.8 – Création des images VO pour permettre l'appariement des images.

L'image de la face vue depuis l'extérieur de la pièce est obtenue par calcul du VO depuis la carte de profondeur. Pour obtenir l'image de la face vue depuis l'intérieur de la pièce, il faut calculer 1-VO puis donner un effet miroir par rapport à l'axe Y.

### Mise en correspondance

Pour chaque image VO obtenue à l'étape précédente, une liste de points d'intérêt et leur descripteur associé est calculé à l'aide des fonctions OpenCV [Bradski, 2000] disponibles.

**Appariement :** Sur la base des listes de points d'intérêt et de descripteurs, les faces de chaque pièce sont testées avec toutes celles de chaque autre pièce. L'appariement se fait en appliquant la stratégie du premier rapport de voisinage géométriquement incohérent le plus proche symétrique (sym FGINN) proposées par Dmytro Mishkin<sup>3</sup>. Cette stratégie permet dans un premier temps de mettre en correspondance des paires de coordonnées dans le système image entre l'image source et l'image de destination. Ensuite une homographie est calculée sur ces paires de points en utilisant RANSAC pour ne garder que des paires de points ayant une cohérence géométrique dans l'image (Figure 2.9). On récupère en sortie les coordonnées images sources et de destination ainsi que le nombre de correspondance entre les deux images.

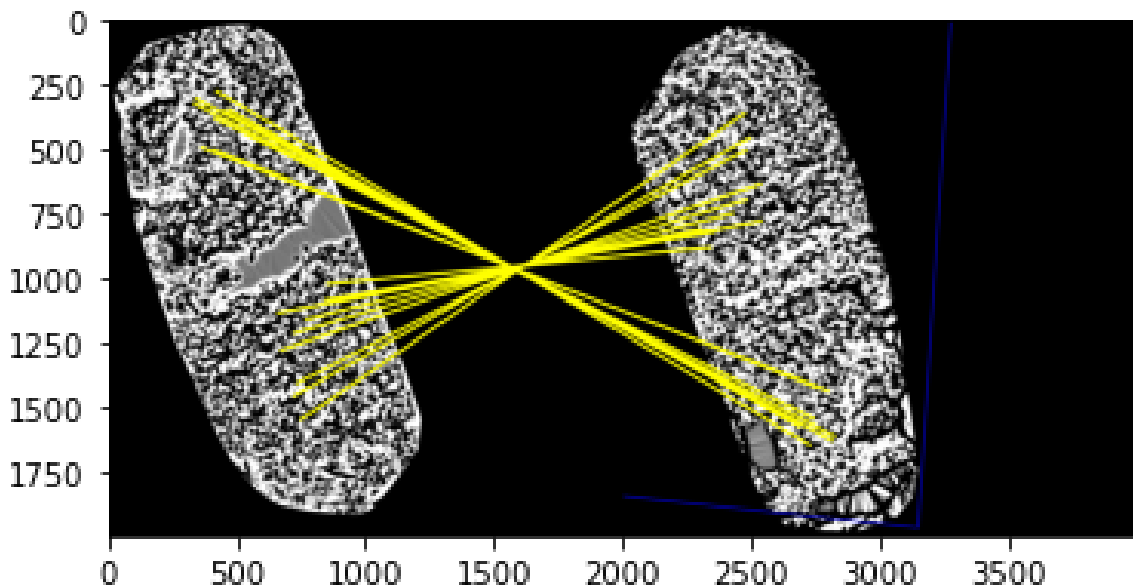


FIGURE 2.9 – Appariement de deux faces avec en jaune les correspondances trouvées entre les points d'intérêts des deux images.

**Transformation :** Pour appliquer au nuage source la transformation qui correspond à l'appariement trouvé ci-dessus, il faut dans un premier temps retransformer les coordonnées du système image en coordonnées 3D correspondant au nuage de points des faces correspondantes. Ensuite, les paramètres de rotation  $\Omega$ ,  $\Phi$  et  $\kappa$  ainsi que ceux de translation  $t_X$ ,  $t_Y$  et  $t_Z$  sont calculés via les moindres carrés entre les paires de coordonnées 3D des nuages à appairer. Un critère de nombre maximum d'itérations ainsi qu'une valeur de stop sur l'accroissement des paramètres à ajuster sont paramétrables. Finalement, pour amener le nuage source sur le nuage de destination, on lui applique, la rotation vers la face source, la transformation issue des moindres carrés et l'inverse de la rotation entre le nuage de destination et la face de destination (Figure 2.10).

3. [https://local-features-tutorial.github.io/pdfs/Local\\_features\\_from\\_paper\\_to\\_practice.pdf](https://local-features-tutorial.github.io/pdfs/Local_features_from_paper_to_practice.pdf)

$$\text{Nuage transformé} = \text{Nuage source} @ R_{\text{src}} @ T_{\text{src-dst}} @ R_{\text{dst}}^{-1}$$

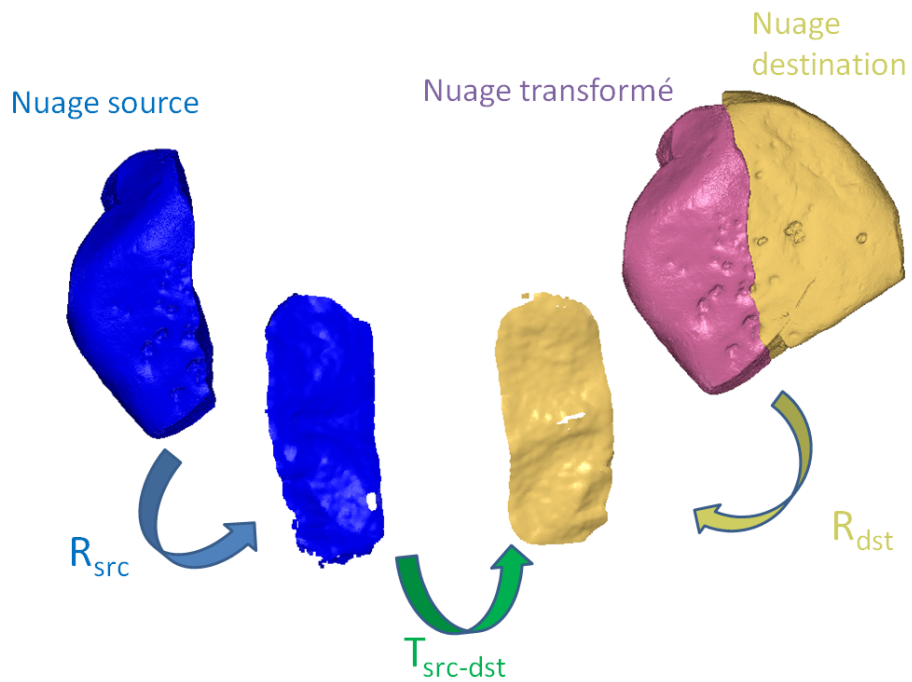


FIGURE 2.10 – Transformation pour amener le nuage source vers le nuage de destination.

Comme la Figure 2.11 le montre, le résultat de calage grossier entre face source et face de destination donne des résultats excellents. deux-tiers des points ont une précision de calage inférieure à 0,2mm et 95% inférieure à 0,6mm pour un nuage initial avec une résolution de 0,3mm. Au vu des résultats, un calage fin n'est pas nécessaire sur la transformation locale.

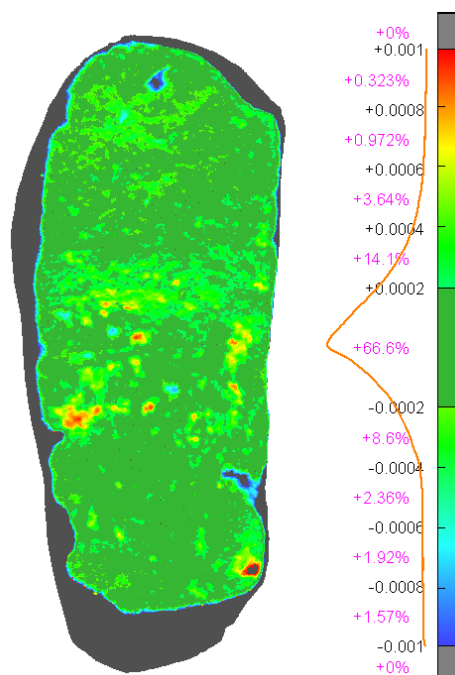


FIGURE 2.11 – Inspection 3D nuage à maillage entre le nuage source transformé et le nuage destination.

### 2.3.2 Appariement à l'aide d'ISS

#### Appariement de nuages de points

Cette méthode doit permettre l'appariement nuage à nuage des différents blocs d'un remontage dans le but de déterminer la transformation entre les deux blocs. Dans un premier temps, le nuage de points (Figure 2.12a) est traité pour extraire des points d'intérêt ayant une valeur de saillie maximale à l'aide de l'ISS. Ces points forment un nuage de points d'intérêt (Figure 2.12b). Une fois les points d'intérêts définis, une mise en correspondance des deux nuages est effectuée à l'aide d'un filtre RANSAC permettant de trouver la transformation ayant une correspondance entre le plus grand nombre de points d'intérêt des deux nuages (Figure 2.13a). Le résultat espéré étant un calage des deux faces appariables (Figure 2.13b).

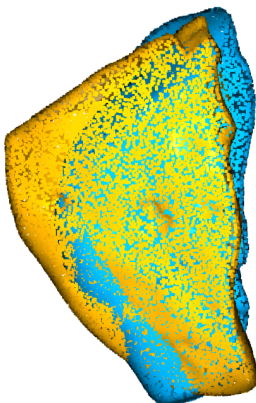


(a) Nuage de point complet avec 1'500'002 points.

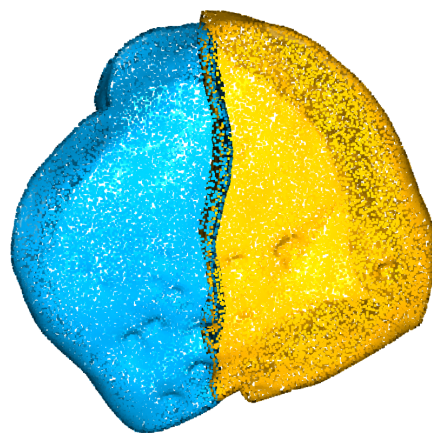


(b) Nuage de points d'intérêt avec 25'561 points.

FIGURE 2.12 – Détection des points d'intérêt du nuage.



(a) Résultat du calage grossier obtenu entre les deux nuages de points.



(b) Résultat de calage correct espéré pour le remontage des deux blocs.

FIGURE 2.13 – Calage grossier à l'aide des points d'intérêt.



### Appariement de faces

Comme on peut le voir sur la Figure 2.13a, le calage grossier effectué sur les nuages superpose les deux nuages car c'est la solution de calage avec le plus grand nombre de correspondances correctes sur l'ensemble des deux nuages. Puisque le but du remontage d'objet est d'apparier des faces de chaque bloc ensemble, il est possible de procéder de la même manière en ayant préalablement découpé le nuage de points en faces (Figure 2.14a). Une fois le découpage effectué, l'appariement peut se faire sur les faces des deux blocs (Figure 2.14b).

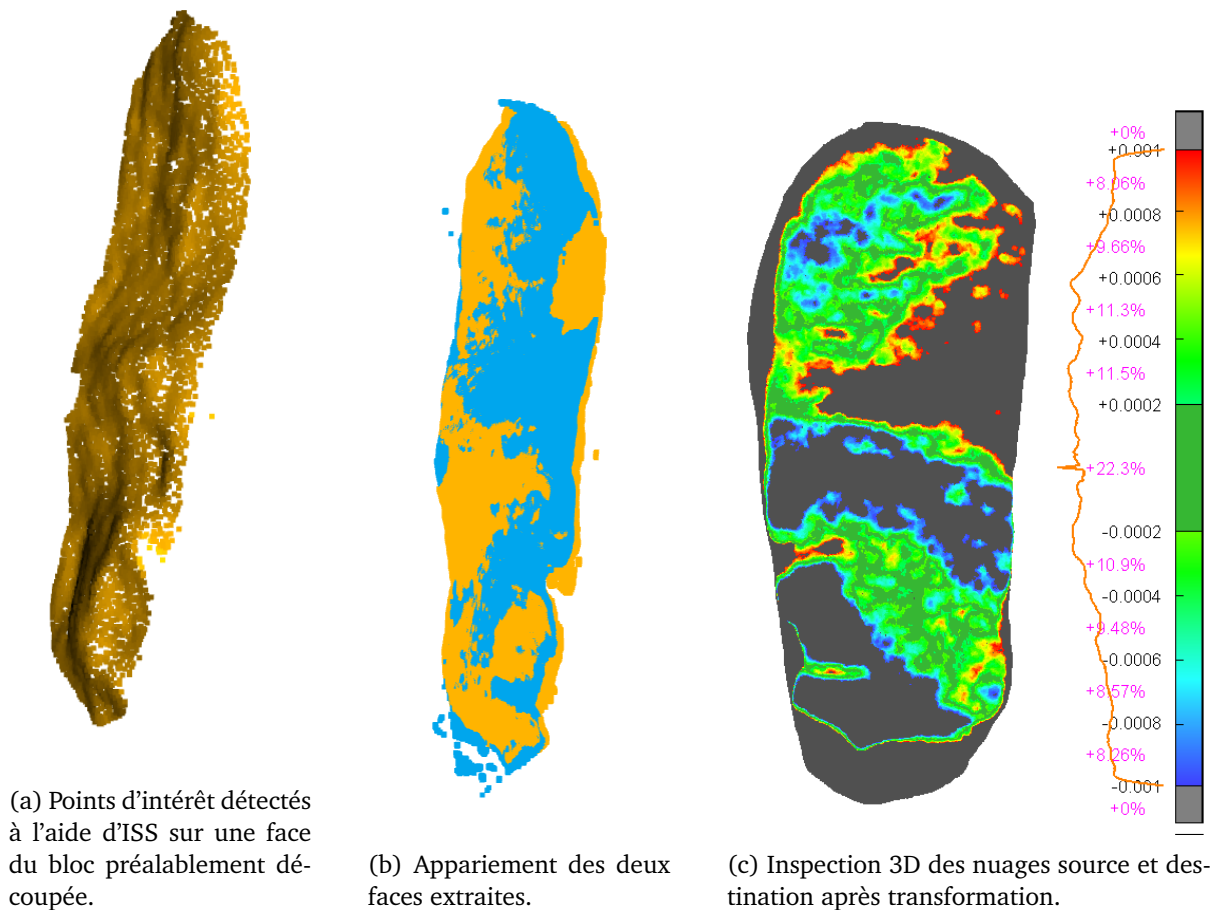
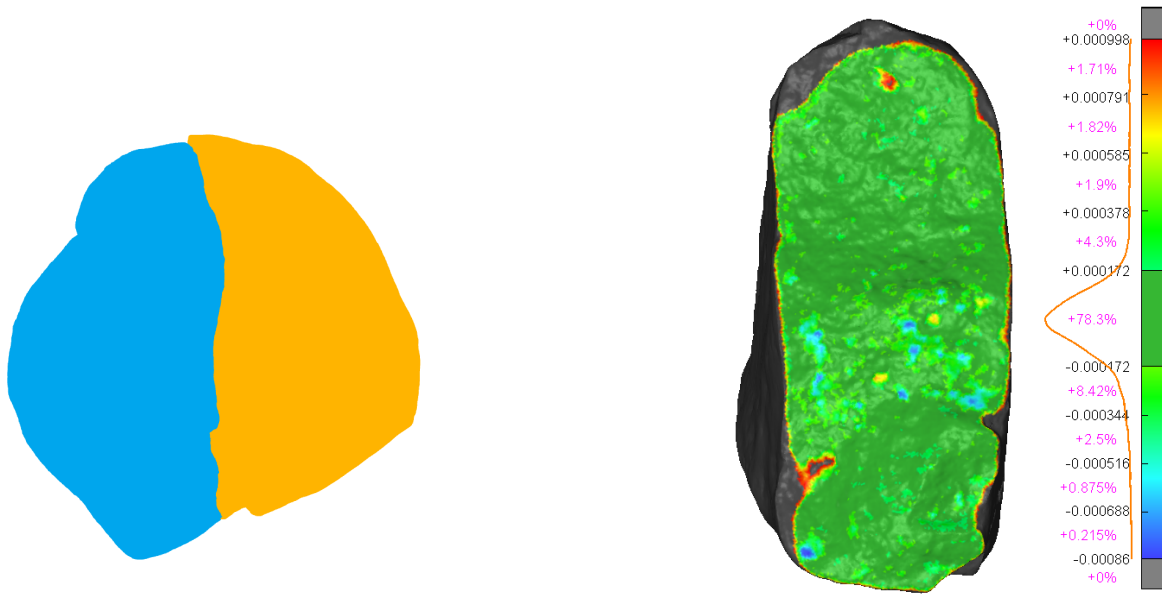


FIGURE 2.14 – Calage grossier de faces à l'aide des points d'intérêt.

La Figure 2.14c montre que les résultats du calage grossier ne sont pas optimaux. Pour améliorer la précision du calage, un calage fin à l'aide d'ICP peut être lancé pour améliorer les résultats de la transformation déterminée ci-dessus.

## Calage fin

Le calage grossier effectué permet d'avoir une transformation approchée entre les deux faces. A partir de celle-ci, un calage fin ICP peut être lancé entre les faces source et destination. L'objectif du calage fin entre les faces est d'améliorer la précision de la transformation avant de l'appliquer au nuage complet. Finalement, la transformation obtenue entre les faces peut être appliquée au nuage source (Figure 2.15a).



(a) Remontage des deux nuages é l'aide de la transformation calculée par l'ICP

(b) Inspection 3D des nuages source et destination après transformation.

FIGURE 2.15 – Calage fin ICP

Le résultat du calage fin visible dans la figure 2.15b est nettement meilleur que celui du calage grossier initial. La précision de calage est inférieure à 0,2mm.

## 2.4 Filtrage des solutions

Avec les solutions de la section 2.3, le remontage de faces qui s'apparient correctement est possible. Il reste maintenant à voir comment il est possible de séparer les bons des mauvais appariements proposés par les deux solutions de calage grossier. Pour illustrer ce filtrage, l'ensemble des faces des deux blocs-sièges sont prises en compte.

### 2.4.1 Filtrage des solutions SIFT

Bien qu'un premier filtre ait déjà été appliqué lors du calcul de l'homographie avec RANSAC, de nombreux mauvais appariements sont encore présents dans les solutions. Deux étapes de filtrage peuvent encore être mises en place pour conserver un maximum de bons appariements tout en excluant les mauvais. La première consiste à limiter les solutions à l'aide d'un nombre minimum de correspondances à atteindre entre deux images. La deuxième consiste à limiter le nombre d'itérations possibles du calcul de moindres carrés pour atteindre la valeur de stop sur l'accroissement des paramètres.

### Filtrage sur les correspondances entre images

Sur la base du décompte du nombre de correspondances entre images retenu pour la meilleure homographie, il est possible d'éliminer les solutions au-dessous d'un nombre minimum de correspondances (Figure 2.16). Plus le critère de sélection est haut, plus la probabilité de n'avoir que des bons appariements est grande, mais les chances que de bons appariements ne soient pas détectés augmentent aussi. Le choix est fait d'utiliser un critère de sélection assez bas sur ce premier filtre pour augmenter les chances de récupérer de bons appariements et de mettre en place un deuxième filtre pour éliminer les mauvais appariements qui ont passé le premier. Le paramètre du nombre de correspondances fera l'objet d'un test dans le Chapitre 3.

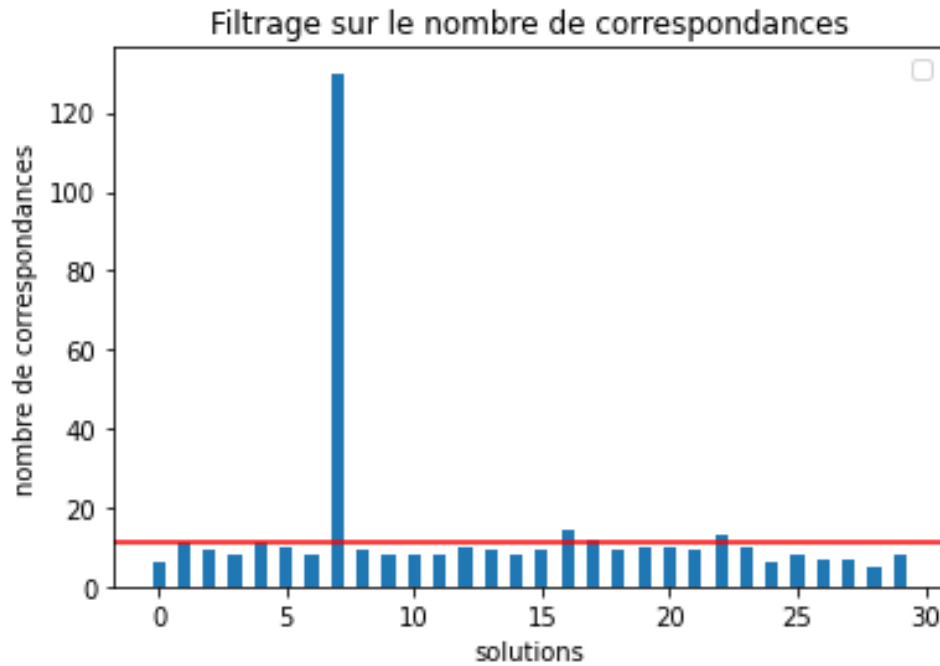


FIGURE 2.16 – Résultat de filtrage des solutions d'appariement entre chaque face du bloc 1 et chaque face des deux blocs-sièges de l'Annexe A.2, soit 5 faces et 6 faces donc 30 solutions possibles.

### Filtrage sur les itérations des moindres carrés

Pour les solutions restantes qui n'ont pas été filtrées par le nombre de correspondances entre images, il est possible de les filtrer une seconde fois sur le nombre d'itérations que le calcul de moindres carrés met à atteindre la valeur de stop sur l'accroissement des paramètres. Seules les solutions présentant un nombre d'itérations inférieur à la limite fixée seront conservées comme appariements utilisables pour le remontage (Figure 2.17).

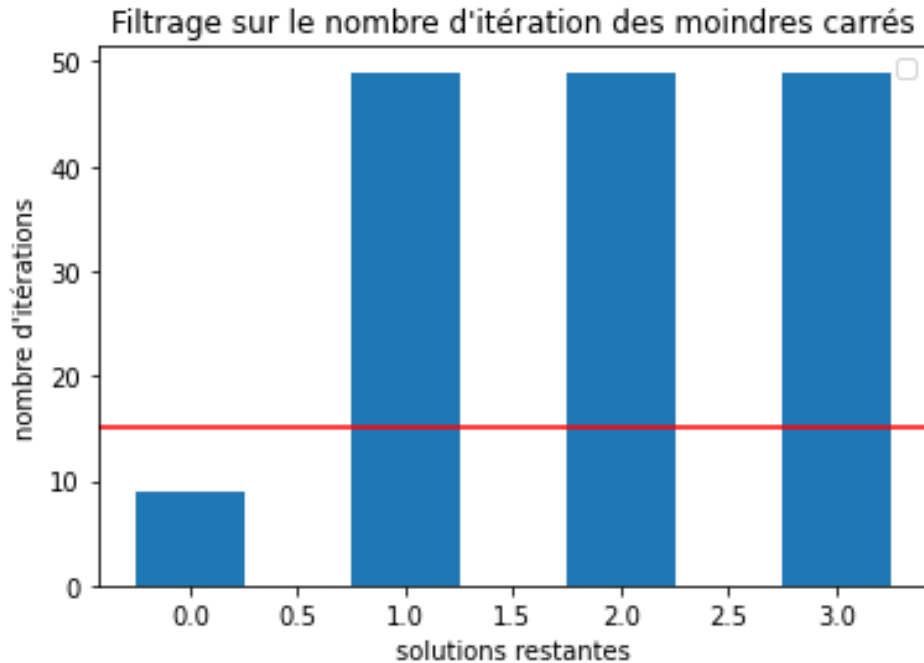


FIGURE 2.17 – Résultat de filtrage des solutions restantes après le premier tri. La solution sous le seuil est correcte, les autres sont fausses.

### 2.4.2 Filtrage des solutions ISS

Le résultat du calage grossier des nuages de points via Open3D donne plusieurs informations qui peuvent être utiles pour séparer les bonnes des mauvaises solutions. Les trois indicateurs sont : le nombre de correspondances établies entre le nuage de points source et de destination, la zone de chevauchement (nombre de correspondances / nombre de points dans la source) et l'erreur moyenne quadratique de toutes les correspondances.

### Nombre de correspondances

Le nombre de correspondances trouvées entre face source et face de destination (Figure 2.18) ne peut être utilisé seul comme filtre des solutions, car il est dépendant de la taille des faces appariées. En revanche, il peut être utilisé dans le critère de la zone de chevauchement.

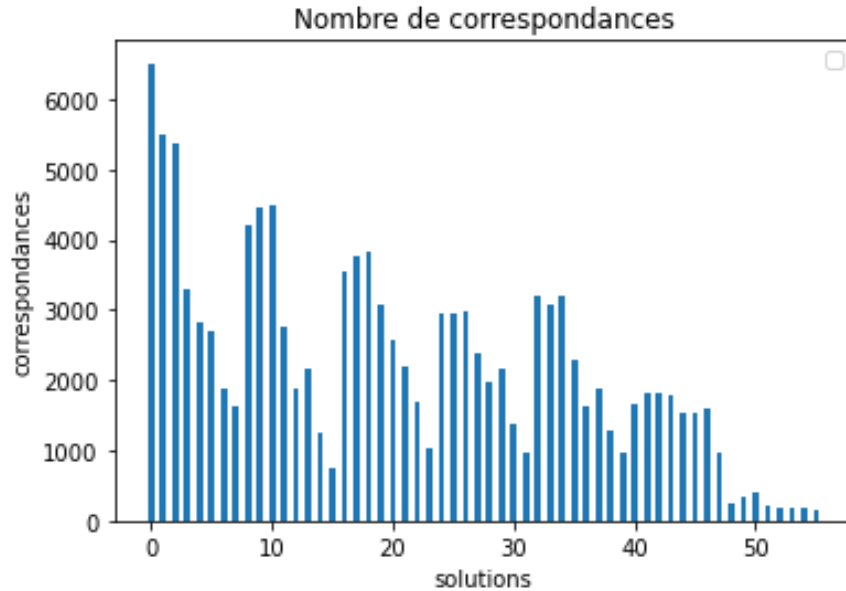


FIGURE 2.18 – Résultat de filtrage des solutions sur le nombre de correspondances.

### Zone de chevauchement

La zone de chevauchement (Figure 2.19) serait utilisable comme critère de tri si on pouvait s'assurer que chaque bonne solution du remontage est un appariement entre une face entière de deux pièces. Or dans les remontages lithiques, les faces peuvent se chevaucher partiellement ce qui rend ce critère inutilisable en l'état. Pour espérer pouvoir utiliser ce critère, il faudrait le pondérer par la surface de la zone de contact pour obtenir une densité de correspondance sur la surface de contact.

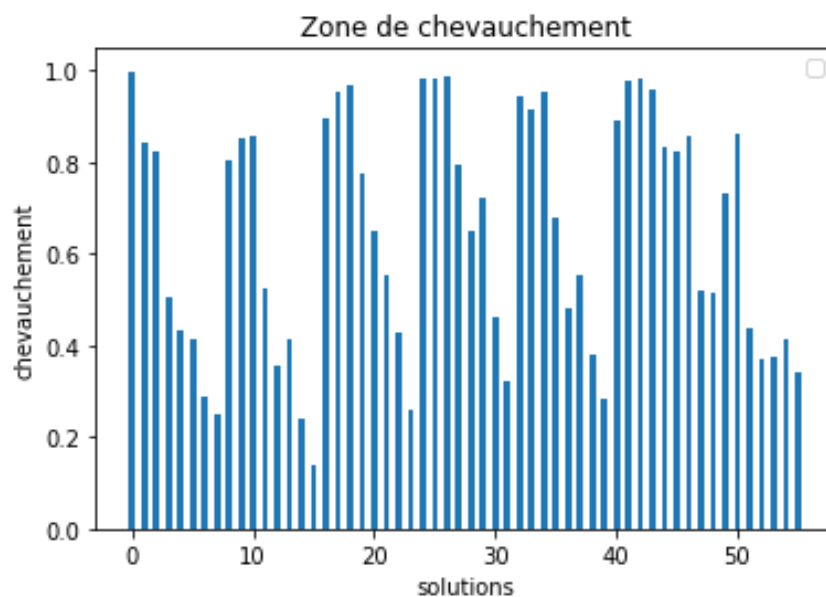


FIGURE 2.19 – Résultat de filtrage des solutions sur la zone de chevauchement.

### Erreur moyenne quadratique des correspondances

L'erreur moyenne quadratique (Figure 2.20) peut filtrer les bonnes solutions, mais l'écart entre les bonnes et mauvaises solutions est très faible du fait que les points ne sont comptés comme correspondances qu'au-dessous du seuil fixé.

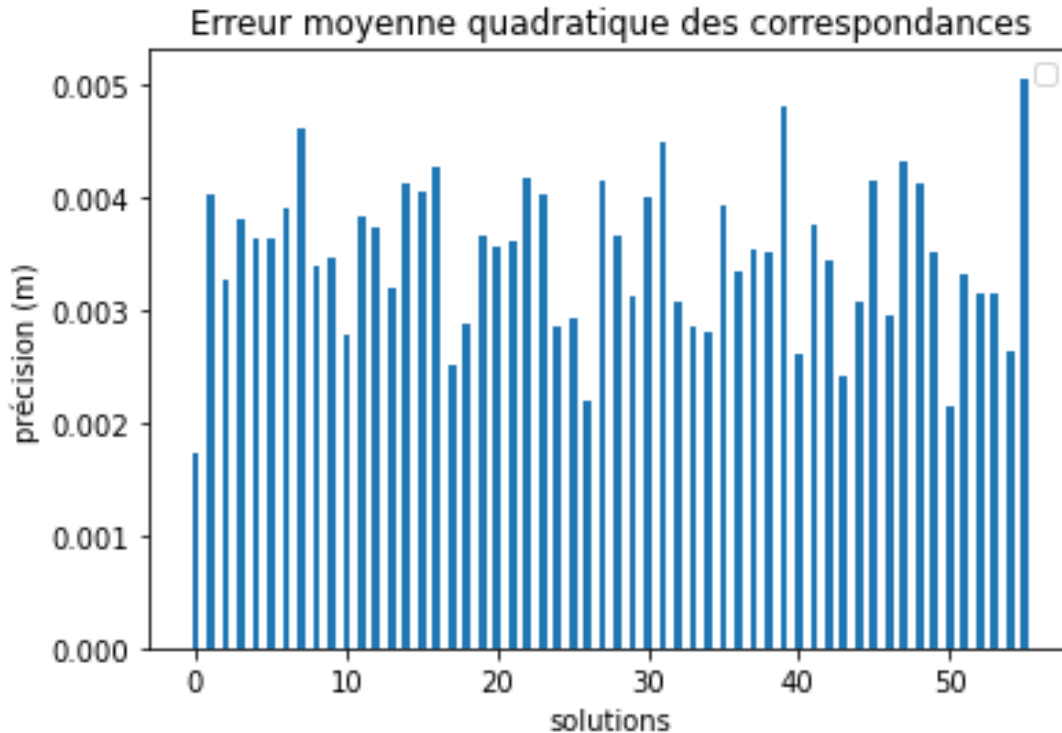


FIGURE 2.20 – Résultat de filtrage des solutions sur l'erreur moyenne quadratique des correspondances.

### Comparaison du filtrage de solutions des deux méthodes d'appariement

Les résultats de filtrage de l'appariement SIFT montrent une différence nette entre la bonne solution et les mauvaises solutions dans les deux filtres successifs. Dans le cas des calages ISS, les indicateurs en sortie du calcul ne permettent pas tous un filtrage et la différence entre bonne et mauvaises solutions est faible. L'appariement à l'aide de SIFT a été retenu pour la suite du travail. La décision est motivée par la précision des calages grossiers obtenus, la capacité à assembler des faces indépendamment du recouvrement de celles-ci et de la possibilité d'appliquer des filtres efficaces pour le tri des bonnes et mauvaises solutions. Pour autant, la solution d'appariement à l'aide d'ISS ne peut pas être invalidée avec les recherches effectuées dans le cadre de ce travail, les recherches n'ayant pas été poussées plus loin pour cette solution.

## 2.5 Remontage de l'objet

A l'issue de la phase de filtrage des solutions, on peut considérer que les appariements restants sont des solutions correctes. Il en ressort une liste avec les nuages d'origine, ceux de destination et les transformations locales pour passer de l'un à l'autre. Dans l'optique de remonter l'objet, les transformations locales doivent être transformées en transformations globales permettant non plus l'appariement par paires, mais le remontage global de l'objet. La première pièce de la liste des pièces appariées est définie comme fixe. Les pièces qui s'apparient directement avec celle fixée se voient appliquer leur transformation locale. Pour les autres pièces, les transformations locales successives permettant une transformation vers la pièce fixe sont imbriquées. Plusieurs morceaux peuvent être créés si les appariements trouvés ne permettent pas de remonter tout l'objet en un seul morceau. En fonction de la qualité du remontage, un calage fin global à l'aide de l'ICP peut encore être appliqué de manière à répartir les erreurs sur l'ensemble des liens du remontage.





## Chapitre 3

# Résultats

Ce chapitre est divisé en quatre parties. La première présente les jeux de données qui seront utilisés pour effectuer les tests. La deuxième expose les tests effectués sur différents paramètres de calcul de manière à optimiser la chaîne de traitement. La troisième déroule la chaîne de traitement globale retenue sur deux jeux de données pour tester la capacité de remontage dans des cas réels. Enfin, la dernière partie se penche sur les limitations de la méthode développée. Le script d'appariement ainsi que les données de base utilisées sont disponibles en annexe numérique. Un "guide utilisateur" et un "guide programmeur" sont disponibles en Annexes B.1 et B.2.

### 3.1 Présentation des jeux de données

Dans ce chapitre, trois jeux de données vont être utilisés : Lokalalei 2C, MIBR16 et JGBO1. Une courte présentation de chaque jeu est proposée ci-dessous.

#### 3.1.1 Lokalalei 2C

Il s'agit ici de maillages des moulages du remontage Lokalalei 2C (Figure 3.1) [Delagnes and Roche, 2005]. Les 13 pièces que compte le jeu proviennent d'un galet de lave. La plus grande pièce mesure 13cm x 5cm x 2cm et la plus petite 4cm x 2cm x 1cm. Les pièces comptent entre 200'000 et 1'000'000 de points pour une résolution de 0,3mm.

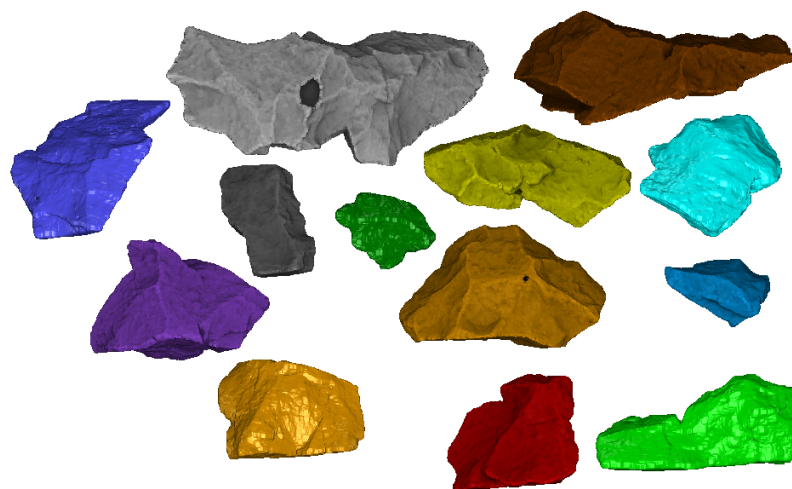


FIGURE 3.1 – Les 13 pièces du remontage Lokalalei 2C étalées.

### 3.1.2 MIBR16

Ce jeu de donnée est un débitage expérimental<sup>1</sup> de 16 pièces en matière siliceuse (Figure 3.2). La plus grande pièce mesure 11cm x 7cm x 4cm et la plus petite 4cm x 2cm x 0,5cm. Les pièces comptent entre 50'000 et 150'000 de points pour une résolution de 0,2mm sur les bord de pièces où il y a une légère rugosité et 0,4mm - 0,5mm lorsque la rugosité est moindre.

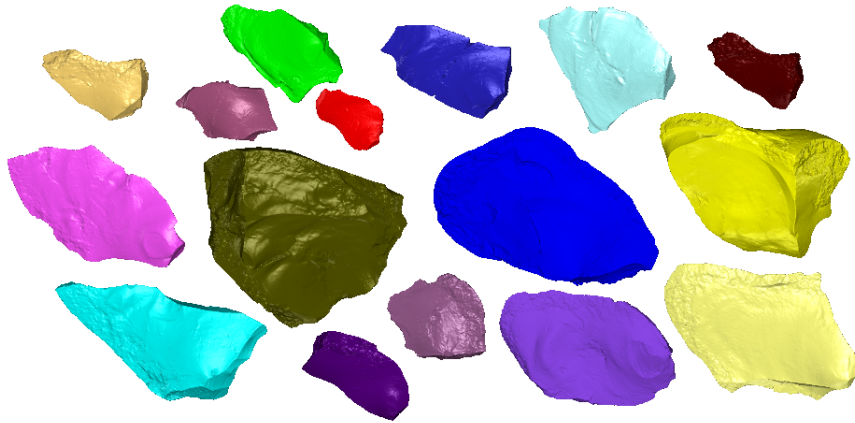


FIGURE 3.2 – Les 16 pièces du remontage MIBR16 étalées.

### 3.1.3 JGBO1

Dans ce jeu de données en matière siliceuse<sup>2</sup>, deux faces qui s'apparient (Figure 3.3a et 3.3b) ont été choisies pour une acquisition photogrammétrique. L'acquisition des deux faces à l'aide d'un appareil photo Sony 24mm a donné lieu à des restitutions à trois résolutions différentes : 0,15mm, 0,3mm et 0,6mm. Ces restitutions donnent des nuages de points à environ 100'000, 50'000 et 25'000 points.



(a) Face A utilisée pour le test.



(b) Face B utilisée pour le test.

FIGURE 3.3 – Les deux faces du jeu JGBO1 utilisées pour les tests.

1. réalisés par Michel Brenet de l'Institut National de Recherches Archéologiques Préventives (INRAP), numérisés par le Max Planck Institute

2. Débitages expérimentaux réalisés dans le cadre de travaux pédagogiques à l'Université de Bordeaux, sous l'encadrement de Jean-Guillaume Bordes, Maître de conférence.

## 3.2 Test des différents paramètres

Afin d'optimiser la chaîne de traitement d'appariement SIFT, des tests sur les paramètres principaux du script ont été effectués pour déterminer quelles valeurs sont les plus à même de permettre l'appariement du maximum de faces correctes. Les premiers tests sont effectués sur la création des faces. Une seconde série de tests est effectuée sur la création des images. Finalement, l'influence des paramètres de filtrage des solutions sera testée.

### 3.2.1 Découpage des faces par propagation des normales

Pour ce découpage de faces, trois paramètres entrent en ligne de compte, la distance maximale de recherche des points voisins, l'angle maximum entre la normal du plan et la normale du point pour l'inclusion du point dans le plan et le nombre minimal de points pour considérer un ensemble de points extrait comme face. La distance maximale est liée à l'espacement des points du nuage et sera donc testée comme étant une fonction de la distance moyenne entre points voisins du nuage. De même, le nombre minimum de points dans un plan est lié au nombre de points total du nuage et sera donc testé comme étant une fonction de ce nombre. Le test a été effectué sur quatre pièces du jeu de données Lokalei 2C visibles dans la Figure 3.4 ci-dessous. Ces pièces ont été sélectionnées de manière à avoir deux grandes pièces assez complexes au découpage et deux petites pièces avec des faces moins grandes dans le but de trouver des paramètres permettant d'extraire l'ensemble des faces de manière à pouvoir les utiliser ensuite dans les appariements.

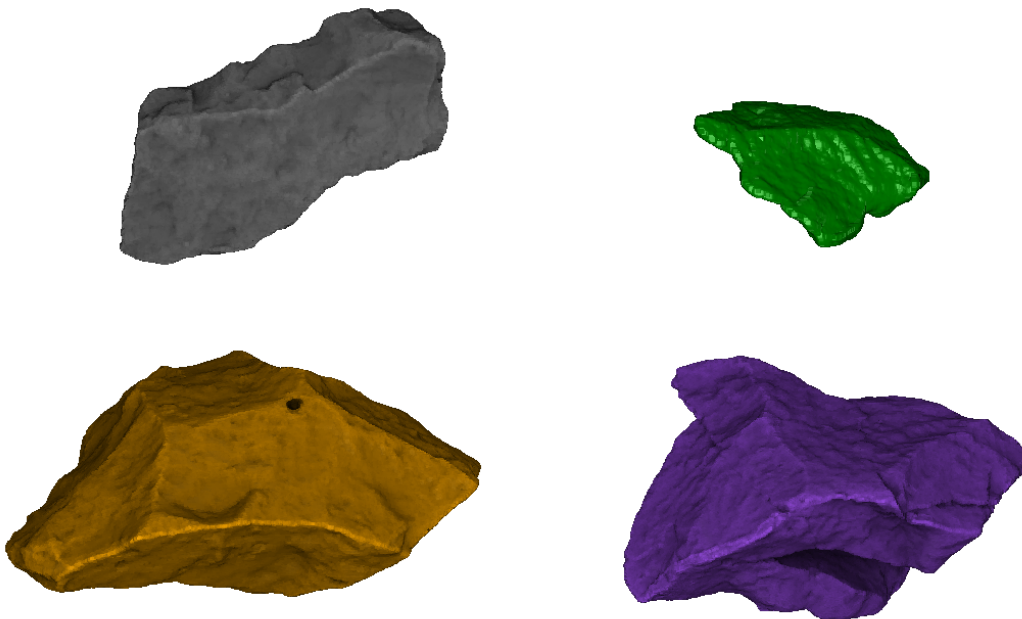


FIGURE 3.4 – Les quatre pièces du Lokalei 2C utilisées pour le test de découpage.

**Résultats :** Le test a été effectué pour une distance maximum entre points de 1X, 2X et 3X la distance moyenne entre points des nuages, un angle maximum de 30°, 40°, 50° et 60° et un nombre de points minimum par face de 1/10, 1/20, 1/30 et 1/40 du nombre moyen de points par nuages. Un tableau récapitulatif des tests est disponible en Annexe A.3. Les solutions les plus adaptées, surlignées en vert, sont les paramètres d'angle de 40° et 50° avec des tailles de faces minimales de 1/40 de la taille moyenne des nuages. Les faces retenues pour la suite des tests sont celles extraites avec les paramètres :

- Distance maximum de recherche = 2 fois la distance moyenne entre points ;
- Angle maximum = 50° ;
- Taille minimum des faces = 1/40 de la taille moyenne des nuages.

Une visualisation 3D des quatre faces avec ces paramètres est disponible en Annexe A.4. Sur la base du test effectué, il n'est pas possible de dire que les paramètres retenus sont optimaux pour n'importe quel jeu de données, mais il s'agit d'une valeur approchée des paramètres à utiliser qui peuvent être ajustés aux besoins de l'utilisateur et aux spécificités du jeu de données à traiter.

**Distance maximum :** Le paramètre de distance maximum ne change rien pour la forme des faces. Cependant il doit être supérieur à la distance entre points du nuage sinon aucune détection de faces n'est possible. Plus il est grand, plus le temps de calcul est long. Le réglage du paramètre à 2 fois la distance moyenne entre points permet une bonne détection des faces sans avoir des temps de calcul trop long. Au besoin, pour des jeux de données dont la distance entre points est irrégulière, le paramètre peut être augmenté.

**Angle maximum :** Le paramètre d'angle maximum influe passablement sur la taille des faces et sur les trous laissés au milieu de celles-ci, l'utilisation d'une valeur entre 40° et 50° pour ce paramètre donne des bons résultats sur le jeu de données test.

**Nombre minimum de points :** Le nombre de points minimum permet de ne conserver que les faces qui auront une chance d'être appariées par la suite. Un nombre de points trop haut va supprimer des faces utiles du jeu de données alors qu'une valeur trop basse conservera des petites faces non-appariables, ce qui augmentera le temps de calcul. Le réglage du paramètre dépendra de la taille des plus petites faces potentiellement appariables dans le jeu de données.

### 3.2.2 Création d'images

La création des images est liée à deux paramètres qui influent sur le résultat d'appariement : la taille du pixel de l'image et le rayon en pixel utilisé pour le VO. Le test a été effectué sur les deux jeux de données Lokalalei 2C et MIBR16.

**Résultats** Sur le premier jeu, les tests ont été effectués avec un rayon de VO de 5, 10, 20 et 30 pixels et une taille d'image entre 200 et 2000 pixels par pas de 200 pixels. Les résultats sont visibles en Figure 3.5. Le nombre d'appariements donnés dans le tableau tient compte des bonnes solutions ayant au minimum 10 correspondances entre les images des deux faces. La meilleure solution donne 15 appariements sur 24<sup>3</sup> possibles dans le jeu de données. Sur le deuxième jeu, les tests ont été adaptés légèrement en fonction des résultats du premier. Les tailles d'images au-dessus de 1400 pixels ont été abandonnées et le pas a été resserré à 100 entre 200 et 600 pour se rendre compte de la variation des solutions avec des tailles d'images plus semblables. Le rayon du VO de 30 pixels a été remplacé par ceux de 3 et 15. Dans ce jeu, 11 appariements sur 37<sup>4</sup> possibles sont trouvés. Les résultats sont visibles en Figure 3.6. Les meilleurs résultats sont obtenus avec les rayons du VO

3. Selon comptage des appariements possibles sur le jeu de données remonté.

4. Selon comptage des appariements possibles sur la partie remontée du jeu de données et estimation pour les pièces non remontées.

les plus faibles, ce qui indique que l'appariement a plutôt tendance à fonctionner sur le traitement local de la géométrie des faces. Les tailles d'images correspondantes à une taille de pixel proche de la résolution native du nuage donnent des meilleurs résultats pour les rayons du VO bas. Lorsque le rayon du VO augmente, la taille du pixel doit diminuer pour obtenir le maximum de solutions possibles avec le rayon donné. Néanmoins, la relation entre les deux n'est pas linéaire comme on pourrait le penser instinctivement.

### LOKALALEI

Taille du pixel	Taille d'image	Rayon VO			
		5	10	20	30
0,00067	200	2	0	0	0
0,00033	400	11	7	1	0
0,00022	600	15	9	5	0
0,00017	800	15	12	6	4
0,00013	1000	12	13	8	4
0,00011	1200	8	13	8	3
0,00010	1400	6	13	11	7
0,00008	1600	3	10	9	6
0,00007	1800	2	10	9	8
0,00007	2000	0	3	9	6

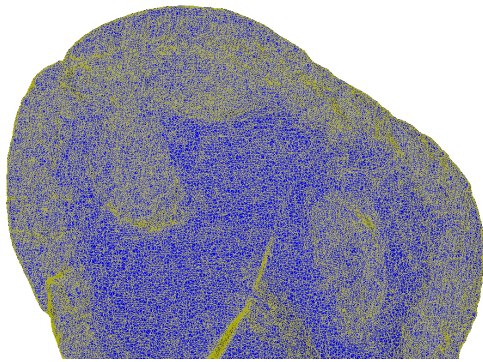
FIGURE 3.5 – Influence des paramètres de rayon du VO et taille d'image sur le nombre de correspondances trouvées pour le jeu de données Lokalalei 2C.

### MIBR16

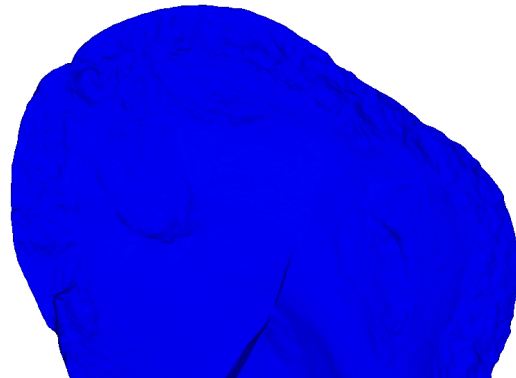
Taille du pixel	Taille d'image	Rayon VO				
		3	5	10	15	20
0,55	200	3	4	2	0	0
0,37	300	7	5	4	3	0
0,28	400	9	10	5	4	3
0,22	500	10	8	6	4	3
0,18	600	10	11	8	5	4
0,14	800	7	11	8	7	5
0,11	1000	4	8	9	8	5
0,09	1200	1	6	8	6	6
0,08	1400	0	3	7	9	6

FIGURE 3.6 – Influence des paramètres de rayon du VO et taille d'image sur le nombre de correspondances trouvées pour le jeu de données MIBR16.

**Taille du pixel** La taille du pixel doit être déterminée par rapport à la résolution du nuage de point. Pour le jeu de données Lokalalei 2C, la résolution du nuage est de 0,3mm. Dans le jeu de données MIBR16, la résolution moyenne est de 0,3mm mais elle n'est pas uniforme sur l'ensemble des nuages et à l'intérieur de chaque nuage. La résolution est de l'ordre de 0,2mm sur l'extérieur des faces où la rugosité est un peu plus marquée et s'approche des 0,4mm sur les parties plus lisses. La comparaison des Figures 3.7a et 3.7b, montre la différence de résolution en fonction de la rugosité de la pièce. L'utilisation d'une taille de pixel autour de 1 à 2 fois la résolution du nuage semble être la meilleure solution au vu des tests effectués.



(a) Pièce du jeu de données MIBR16 avec maillage triangulaire passant par tous les points. Les triangles sont mis en évidence en jaune.



(b) Pièce du jeu de données MIBR16 avec maillage triangulaire passant par tous les points.

FIGURE 3.7 – Visualisation de la différence de résolution sur les pièces du jeu MIBR16.

**Rayon du VO** Le rayon utilisé pour le VO a également une grande influence dans le nombre de correspondances trouvées. Plus le rayon est grand et plus les chances d'inclure des pixels qui ne correspondent pas entre les deux images augmentent. Il faut compter également que les bords de la pièce jusqu'à une fois le rayon du VO ne sont pas utilisables pour l'appariement car les résultats du VO seront faussés dans cette zone.

### 3.2.3 Filtrage des solutions

Le dernier paramètre important ayant fait l'objet d'un test est le nombre de correspondances minimum pour considérer l'appariement comme correct. L'objectif est de conserver toutes les bonnes solutions en éliminant les mauvaises. Comme illustré sur la Figure 3.8, les solutions peuvent être divisées en trois catégories sur la base du nombre de correspondances entre images : celles qui ont un nombre de correspondances élevées et qui seront sans doute correctes (zone verte), celles qui ont un nombre de correspondances faibles et qui ont très peu de chances d'être correctes (zone rouge) et celles qui sont entre deux (zone jaune). Cette dernière catégorie est spécialement intéressante car elle permet de récupérer des appariements supplémentaires qui sont importants si elle est traitée correctement. Le premier défi consiste à savoir jusqu'à quelle valeur il est probable de récupérer de bons appariements. Pour déterminer cette limite un test est effectué sur les mêmes bases que le précédent en sélectionnant la plage comptabilisant le plus de bonnes solutions, soit des tailles d'images entre 600 et 1000 pixels et un rayon de VO entre 5 et 20.

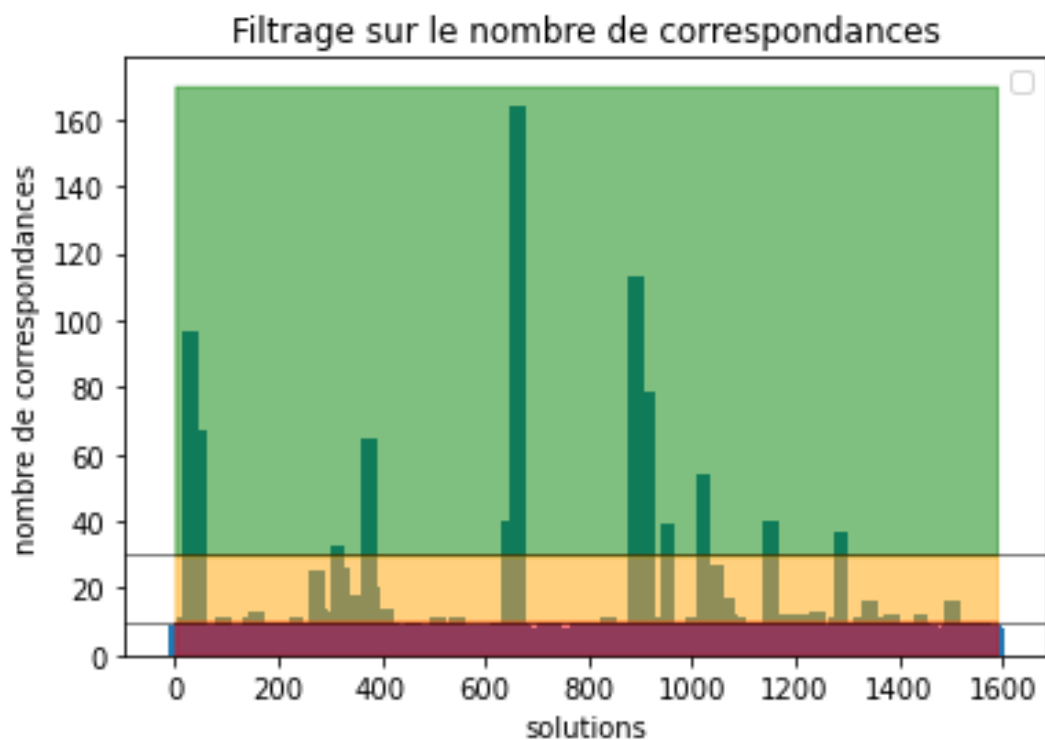


FIGURE 3.8 – Les solutions de la section verte sont correctes, celles de la section jaune doivent être validées par un deuxième filtrage, celles de la section rouge n'ont pas été étudiées car les chances d'y trouver des solutions correctes sont très faibles.

**Résultats** Les appariements corrects et incorrects qui ont passés le test avec 10 et 15 correspondances minimum sont répertoriés dans les Figures 3.9 et 3.10 qui suivent. En plus du nombre de correspondances, le filtre sur le nombre d'itération élimine les solutions qui ne convergent pas avant 50 itérations de la même manière pour les deux tests. Comme on peut le voir sur les figures, la modification du seuil du nombre de correspondances minimum influe fortement sur le nombre de solutions en sortie. Pour un seuil qui passe de 10 à 15 correspondances, 73 mauvaises solutions sur 77, soit 95% sont supprimées contre uniquement 3% de bonnes solutions (3/95). De plus les bonnes solutions supprimées font partie des combinaisons qui en avaient déjà le moins. La valeur de 15 correspondances est utilisable pour filtrer efficacement ce jeu de données.

## LOKALALEI

10 correspondances minimum

Taille du pixel (m)	Taille d'image (pix)	Rayon VO					
		5		10		20	
		JUSTE	FAUX	JUSTE	FAUX	JUSTE	FAUX
0,00022	600	15	6	9	3	5	4
0,00017	800	15	9	12	4	6	11
0,00013	1000	12	21	13	7	8	12

FIGURE 3.9 – Nombre de solutions justes et fausses pour un seuil fixé à 10 correspondances entre images.

## LOKALALEI

15 correspondances minimum

Taille du pixel (m)	Taille d'image (pix)	Rayon VO					
		5		10		20	
		JUSTE	FAUX	JUSTE	FAUX	JUSTE	FAUX
0,00022	600	15	0	8	0	3	0
0,00017	800	15	1	12	0	6	0
0,00013	1000	12	2	13	0	8	1

FIGURE 3.10 – Nombre de solutions justes et fausses pour un seuil fixé à 15 correspondances entre images.

### 3.2.4 Densité du nuage de points

En plus des paramètres de traitement, un autre facteur qui dépend cette fois de l'acquisition entre en ligne de compte, la résolution du nuage de point. Pour tester son influence, les deux faces du jeu de données JGBO1, avec leurs trois résolutions de nuage de points ont été utilisées. Le test consiste à voir avec quels paramètres et combien de fois il est possible d'apparier ces faces en fonction des différentes résolutions.



**Résultats** Les tests montrent comme attendu qu'en diminuant le nombre de points dans les faces, les appariements deviennent plus difficilement détectables (Figures 3.11, 3.12 et 3.13). Néanmoins, le test effectué avec une demi résolution donne encore 13 possibilités d'appariement et serait donc largement suffisant. Il serait intéressant de pouvoir produire ce test à plus grande échelle avec des faces de rugosité différente et des acquisitions à des résolutions différentes. Ce test permettrait de déterminer la résolution nécessaire à l'acquisition pour pouvoir appairer les objets en fonction de leur rugosité.

**JGBO1**

Résolution = 0,00015		Rayon VO								
Taille du pixel (m)	Taille d'image	2	3	4	5	10	15	20	25	30
0,00058	100	0	1	0	0	0	0	0	0	0
0,00029	200	0	1	1	1	0	0	0	0	0
0,00019	300	0	1	1	1	1	0	0	0	0
0,00015	400	1	0	0	0	1	1	1	0	0
0,00012	500	0	0	0	0	1	1	1	0	0
0,00010	600	1	0	0	0	1	1	1	1	1
0,00007	800	0	0	0	0	0	1	1	1	0

Appariement détecté 24 fois

FIGURE 3.11 – L'appariement entre les deux faces a été trouvé 24 fois pour le nuage à pleine résolution.

**JGBO1**

Résolution = 0,0003		Rayon VO								
Taille du pixel (m)	Taille d'image (pix)	2	3	4	5	10	15	20	25	30
0,00058	100	0	0	0	0	0	0	0	0	0
0,00029	200	0	1	1	1	1	0	0	0	0
0,00019	300	0	0	1	1	1	1	0	0	0
0,00015	400	0	0	0	0	1	1	0	0	0
0,00012	500	0	0	0	0	1	1	0	0	0
0,00010	600	0	0	0	0	0	1	0	0	0
0,00007	800	0	0	0	0	0	0	0	0	0

Appariement détecté 13 fois

FIGURE 3.12 – L'appariement entre les deux faces a été trouvé 13 fois pour le nuage à une demi résolution.

**JGBO1**

Résolution = 0,0006		Rayon VO								
Taille du pixel (m)	Taille d'image (pix)	2	3	4	5	10	15	20	25	30
0,00058	100	0	0	0	0	0	0	0	0	0
0,00029	200	0	0	1	1	0	0	0	0	0
0,00019	300	0	0	0	1	0	0	0	0	0
0,00015	400	0	0	0	0	0	0	0	0	0
0,00012	500	0	0	0	0	1	0	0	0	0
0,00010	600	0	0	0	0	0	0	0	0	0
0,00007	800	0	0	0	0	0	0	0	0	0

Appariement détecté 4 fois

FIGURE 3.13 – L'appariement entre les deux faces a été trouvé 4 fois pour le nuage à un quart de résolution.

### 3.3 Déroulement de la chaîne de traitement

Les paramètres ayant pu être testés individuellement, la chaîne de traitement dans son ensemble va être testée sur les deux jeux de données Lokalalei 2C et MIBR16. Elle sera également testée sur les quarts jeux de données présentés dans ce travail, soit : Lokalalei 2C, MIBR16, JGBO1 et les deux blocs-sièges dans le but de tester le remontage multi-objets. L'objectif est de voir s'il est possible de les remonter les jeux de données automatiquement et entièrement. Le test du remontage multi-objets aura également pour objectif de voir si des faux appariements supplémentaires ne sont pas introduits en augmentant le nombre de pièces.

**Lokalalei 2C** Pour ce jeu de données, le remontage complet et correct (Figure 3.14) est exporté après 38 min<sup>5</sup>. Un détail chiffré est disponible à la Figure 3.15. Le remontage complet est disponible en visualisation 3D à l'annexe A.5.

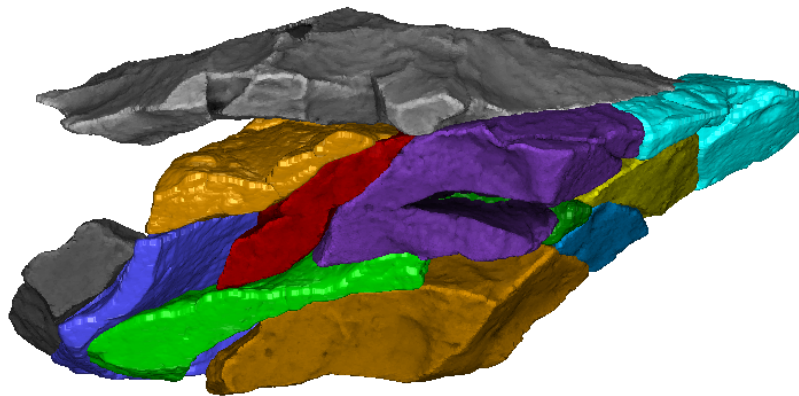


FIGURE 3.14 – Remontage du Lokalalei 2C.

	Lokalalei 2C	MIBR16
<b>Nombre de pièces</b>	13	16
<b>Taille des données</b>	459 Mo	88 Mo
<b>Nombre de faces</b>	59	42
<b>Appariements à calculer</b>	1589	985
<b>Solutions après 1 test</b>	26	25
<b>Solutions après 2 test</b>	15	9
<b>Solutions supplémentaires</b>	0	2
<b>Nombre de pièces remontées</b>	13 (100%)	11 (69%)
<b>Temps de calcul en secondes</b>		
<b>Import des données</b>	11	3
<b>Extraction des faces</b>	2009	585
<b>Préparation des images</b>	164	67
<b>Appariement</b>	116	107
<b>Moindres carrés</b>	11	3
<b>Transformation des nuages</b>	1	0

FIGURE 3.15 – Statistiques sur le traitement des jeux de données Lokalalei 2C et MIBR16.

5. Calcul sur une machine avec processeur Intel Core i7 / 16Go de RAM.

**MIBR16** Pour ce jeu de données, un remontage de 9 pièces est exporté après 13 min<sup>6</sup>. En comparant les liens trouvés avec le test effectué au chapitre 3.2.2, il est possible de voir que deux liens supplémentaires peuvent être trouvés en utilisant les paramètres de création d'images de 400 pixels pour la taille de l'image. En ajoutant ces deux liens au calcul, on peut arriver à un remontage de 11 pièces sur les 16 que compte le jeu (Figure 3.16). Les faces qui n'ont pas pu être remontées sont visibles en Figure 3.17. Un détail chiffré est disponible à la Figure 3.15. Le remontage complet est disponible en visualisation 3D à l'annexe A.6.

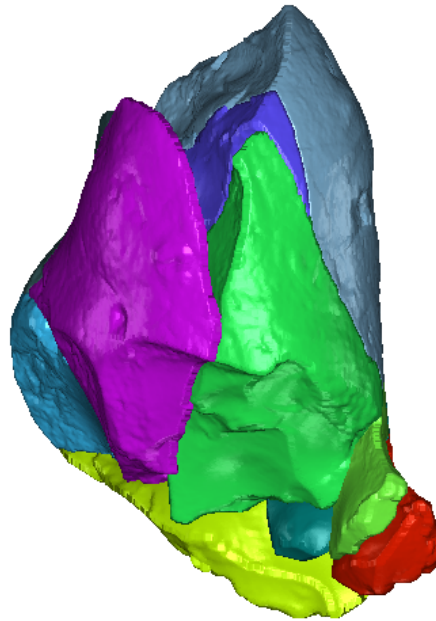


FIGURE 3.16 – Remontage du MIBR16.



FIGURE 3.17 – Pièces restantes après le remontage.

---

6. Calcul sur une machine avec processeur Intel Core i7 / 16Go de RAM.

**MIBR16 / analyse :** Les pièces restantes qui n'ont pas pu être remontées sont les plus petites pièces, mais également celles qui ont le moins de rugosité. Parmi les pièces qui ont pu être appariées, le constat est le même : les correspondances sont trouvées sur les zones les plus rugueuses (Figure 3.18).

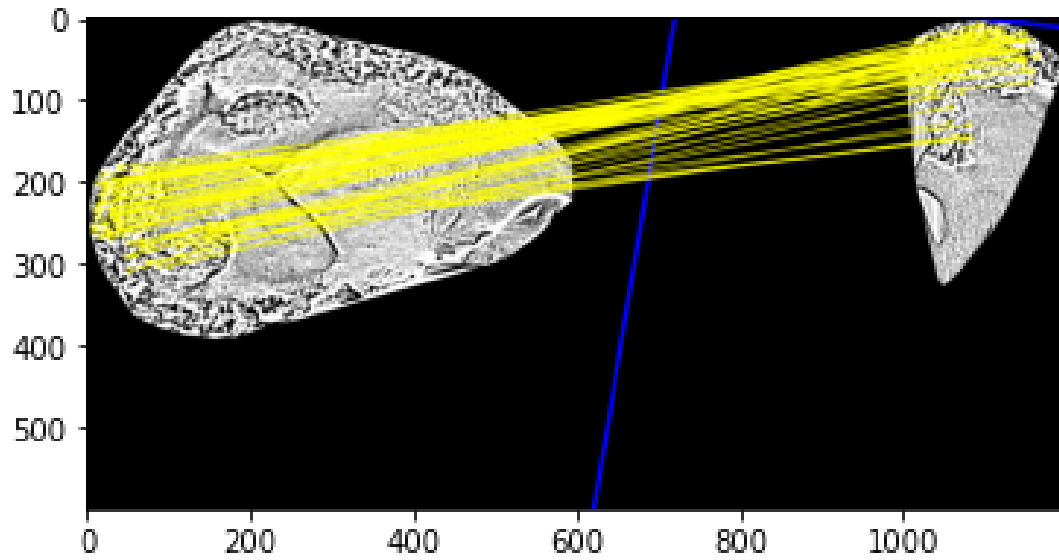


FIGURE 3.18 – Correspondances trouvées pour l'appariement entre deux pièces du MIBR16.

**Remontage multi-objets :** En lançant le calcul sur les quatre jeux de données vus précédemment dans ce travail avec les paramètres suivants :

- Résolution du nuage = 0,3mm (moyenne sur les différent jeux de données)
- Nombre de points minimum par nuage = 150'000 (pris sur les plus petits nuages)
- Nombre de voisins utilisés pour le calcul des normales = 5
- Angle maximal = 50°
- Taille du pixel = 0,3mm (idem résolution du nuage)
- Valeur du rayon du VO = 5
- Nombre minimum de correspondances = 15
- Nombre maximum d'itérations = 25

Les nuages exportés de la Figure 3.19 sortent après 2h09<sup>7</sup>. 90% de ce temps de calcul est dédié au découpage des plans. Les deux blocs-sièges ainsi que les plus grosses pièces du Lokalalei étant particulièrement lourde, elle ralentissent fortement cette étape. 4% du temps total de calcul est ensuite dédié à la création des images. L'appariement représente 5% du temps de calcul pour 9208 appariements à tester et le restes des opération se partage le dernier 1%. Au total, 94% du temps de calcul est utilisé sur des étapes qui peuvent être parallélisées. Dans les résultats exportés, les deux blocs-sièges sont correctement recalés. Sept pièces du MIBR16 sont également correctement recalées. Dix pièces du Lokalalei 2C sont recalées en deux morceaux, mais une pièces du MIBR16 s'est également calé avec (pièce en bleu sur la Figure 3.19). Les deux faces du JGBO1 n'ont pas pu être calées. Les résultats légèrement moins bons sur les remontages MIBR16 et Lokalalei 2C sont dus aux paramètres moyennés sur l'ensemble des nuages qui sont légèrement différents de ceux utilisés dans les tests. L'échec du calage des faces du JGBO1 s'explique par le fait que sa plage de détection est légèrement différente de celle des autres nuages.

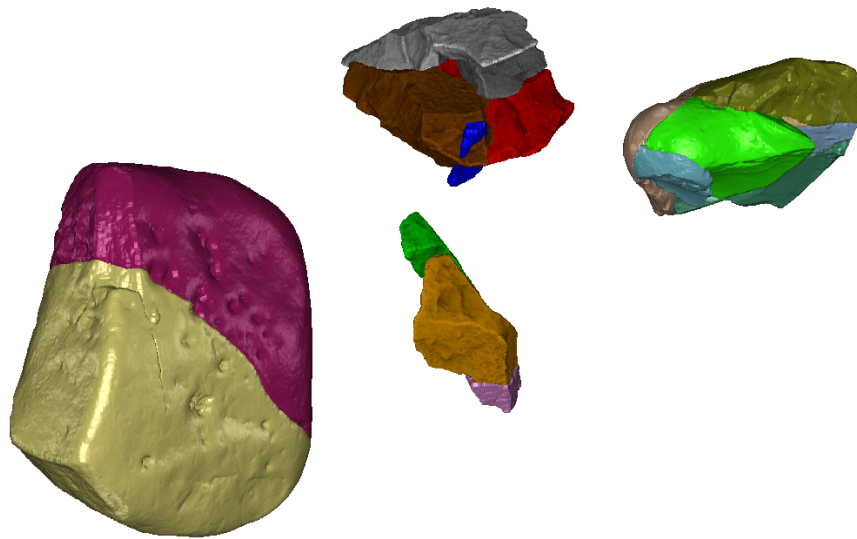


FIGURE 3.19 – Les différents blocs remontés automatiquement par le script.

7. Calcul sur une machine avec processeur Intel Core i7 / 16Go de RAM.

### 3.4 Limitations

En regard aux résultats de la section précédente, l'appariement n'est pas possible pour tous les types de pièces et dans n'importe quelles conditions. La taille relative des pièces ainsi que leur rugosité sont des facteurs limitant pour l'appariement SIFT. De plus, il sera difficile avec le script en l'état de prouver l'absence d'appariement, comme présenté dans les objectifs.

**Taille des pièces :** La taille relative des pièces qui entrent en ligne de compte pour l'appariement est importante pour déterminer quelles pièces pourront être potentiellement appariables. Comme la création des images est basée sur une taille d'image et de pixel fixe pour conserver les proportions entre les pièces, il se peut que des petites pièces ne soient plus appariables. Comme le montre la Figure 3.20, la zone non utilisable au bord de l'image a une taille constante égale au rayon du VO. Cette zone va donc fortement réduire la surface appariaable lorsque les pièces sont petites alors qu'il n'y aura pas de problème pour les plus grandes pièces. Une piste d'amélioration pour minimiser le problème serait d'avoir une meilleure résolution sur les acquisitions de manière à ce que la taille du pixel soit plus petite et donc la zone inutilisable moins grande.

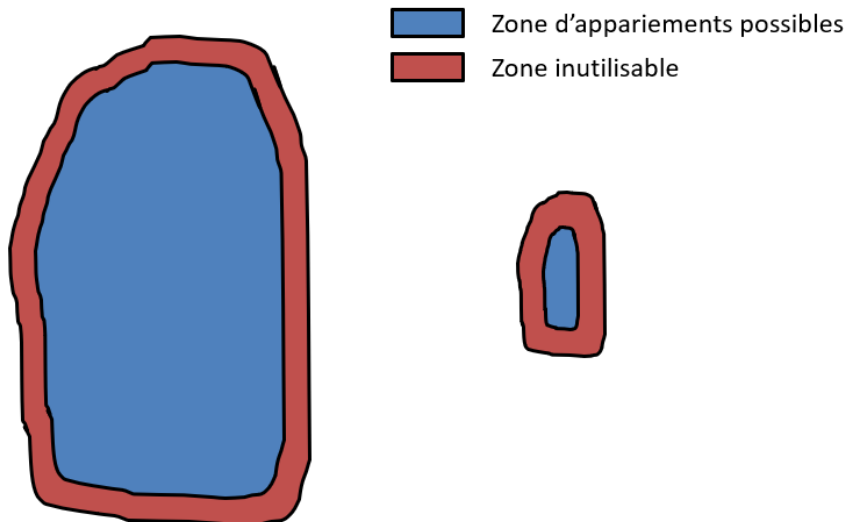
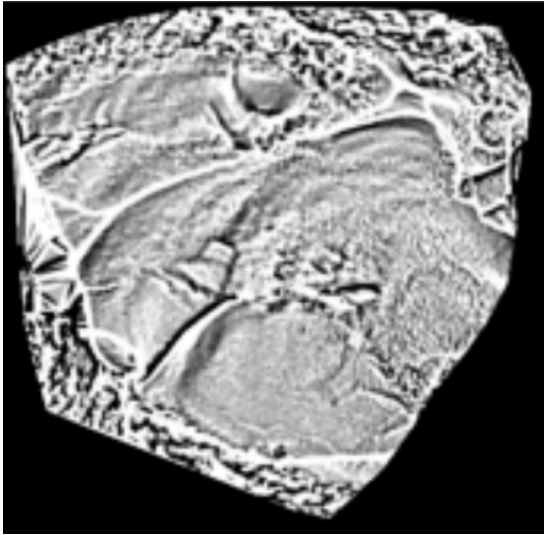
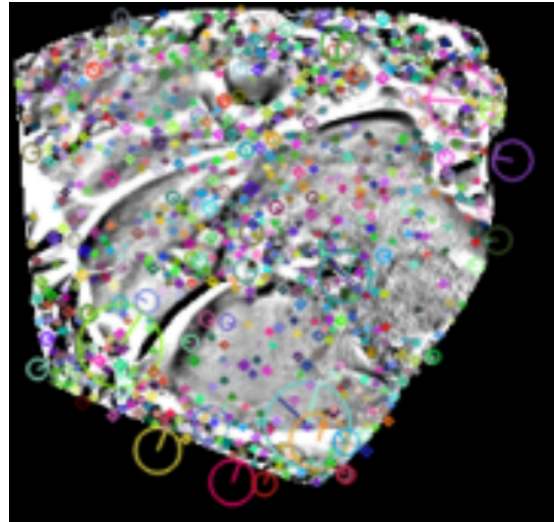


FIGURE 3.20 – Zone utilisable pour l'appariement après traitement de l'image par le VO.

**Rugosité :** Avec le traitement des faces par le VO (Figure 3.21a), les pixels de l'image représentent le volume au dessus de la surface de la sphère du rayon choisi centrée sur le pixel en question. Dans le cas où la surface est lisse, ce volume ne varie pas, ou très peu, entre les pixels voisins. Ce manque de variation a pour conséquence que SIFT ne peut pas trouver de points d'intérêt (Figure 3.21b) sur les zones lisses et donc qu'il ne peut pas y avoir de correspondance dans ces zones.



(a) Vo d'un face du jeu de données MIBR16.



(b) Ajout des points d'intérêts détectés par SIFT (cercles colorés) sur l'image précédente.

FIGURE 3.21 – Répartition des points d'intérêts en fonction de la rugosité.

**Absence d'appariement :** Pour prouver l'absence d'appariement possible dans un jeu de données, il faut être capable de détecter avec une haute fiabilité l'ensemble des appariements probable dans le jeu en question. En reprenant le nombre de liens détectés sur les résultats de remontage de Lokalalei 2C et MIBR16, soit 15/24 et 11/37, ce qui fait une moyenne de 43% d'appariements détectés, le script actuel est loin de pouvoir détecter l'ensemble des appariements.

**Conclusion :** Pour déterminer les possibilités et limites d'utilisations du projet de manière plus fiable, il faudrait procéder à des tests couplés d'acquisition et de détection sur un ensemble de jeu de données ayant des tailles et rugosités et résolutions différentes.





## Conclusion générale

Dans ce travail, il a été étudié la possibilité de faire des remontages d'objets lithique de manière numérique et automatisée. L'objectif a été de faire un inventaire des différentes possibilités d'appariement pouvant permettre d'arriver au résultat escompté. A partir de ces outils pré-existants, des comparaisons et tests ont été effectués pour déterminer une option finale à conserver. Les outils qui ont permis le remontage sont :

- Un découpage des pièces en faces ;
- Une transformation des faces en carte de profondeur ;
- Une mise en évidence des reliefs à l'aide de l'Obscurcissement Volumétrique ;
- Une mise en correspondance SIFT couplée à un tri RANSAC ;
- Une double sélection des solutions au travers des correspondances et des résultats de moindre carrés ;
- Un remontage basé sur les transformations déterminées.

Au final, en se basant uniquement sur la géométrie des objets, il est possible de procéder au remontage d'objets lithiques qui a été testé sur plusieurs jeux de données. Néanmoins, tous les liens possibles entre les différentes pièces ne sont pas forcément retrouvés lors du remontage. Il a été remarqué lors des tests effectués que la rugosité des pièces est un élément limitant pour l'appariement tout comme une trop grande différence de taille entre les pièces à apparier. La résolution des données en entrée a également une grande influence sur la capacité du script à détecter les appariements. En se basant sur ces éléments, il faut conclure en disant que le projet permet d'aider à la recherche de correspondances dans le remontage d'objets lithiques, mais qu'il ne peut pas en l'état être utilisé pour prouver l'absence de correspondance qui se serait avéré être une information tout aussi précieuse.

## Perspectives

Dans le prolongement de ce qui a été fait, d'autres méthodes d'appariements plus spécifiques peuvent permettre l'appariement d'objets moins rugueux. La création d'un descripteur 3D comprenant des informations sur l'orientation du plan local, des informations similaires au VO mais calculé sur le nuage de points ou le maillage, voir un calcul multi-échelles de ce VO. Comme les débitages d'objets siliceux donne souvent des arrêtes franches sur les pièces, un autre axes de recherche consisterait à extraire les arrêtes saillantes des différentes pièces. La mise en correspondance des arrêtes pourrait également constituer une solution pour la mise en correspondance d'objets trop lisse pour la méthode développée dans ce travail.

En partant du principe que pour une utilisation en conditions réelles, les jeux de données à traiter ne serait plus un ensemble de pièces qui remontent, mais un ensemble dans lequel il y a de rares appariements possibles. La robustesse des solutions retrouvées peut être améliorée en utilisant plusieurs tailles d'image ou plusieurs taille de rayon du VO et en recoupant les solutions. Ainsi les solutions détectées plusieurs fois peuvent être validées et celle qui ne sont détectées qu'une fois peuvent être contrôlées manuellement par l'utilisateur via une interface graphique.

Dans l'optique d'une utilisation sur des jeux de données plus grands, la question de la rapidité du script devra également être traitée. Dans un premier temps, deux choses peuvent être faites pour améliorer la vitesse d'exécution : Un codage propre du script dans un langage optimisé pour la rapidité de traitement du type C++ et une parallélisation des opérations de découpage des faces et de construction des images.

## Annexe A

# Annexes techniques

### A.1 Vue 3D de l'extraction de faces par trois points

Les deux PDF 3D qui suivent montrent le découpage en faces des deux blocs-sièges par la méthode des meilleurs plans par trois points.

Comme la visualisation des PDF 3D n'est pas possible avec tous les lecteurs PDF, les nuages de points sont également disponibles dans les annexes numériques.





## **A.2 Vue 3D de l'extraction de faces par propagation des normales**

Les deux PDF 3D qui suivent montrent le découpage en faces des deux blocs-sièges par la méthode de propagation des normales.

Comme la visualisation des PDF 3D n'est pas possible avec tous les lecteurs PDF, les nuages de points sont également disponibles dans les annexes numériques.







### **A.3 Test de découpage en faces par propagation des normales**

Sur la page suivante, le tableau complet de récapitulation du test de découpage en face par propagation des normales est disponible. Il a servi de base pour déterminer les paramètres utilisés pour la suite du projet.

# Test de découpage en faces par propagation de normales

Distance maximum	Angle maximum	Nombre de points minimum	LOKALALEI_1		LOKALALEI_6		LOKALALEI_8		LOKALALEI_12		Temps de traitement du nuage (N°8) (sec)	Somme des faces manquantes	Remarques	
			Nbr face	Faces manquantes	Nbr face	Faces manquantes	Nbr face	Faces manquantes	Nbr face	Faces manquantes				
Approximation manuel du nombre minimum de faces à extraire			5		2		2		7					
d_pts	30 à 60	pts_moy / 10	Distance maximum pas adaptée											Pas de faces extraites
d_pts	30 à 60	pts_moy / 20	Distance maximum pas adaptée											Pas de faces extraites
d_pts	30 à 60	pts_moy / 30	Distance maximum pas adaptée											Pas de faces extraites
d_pts	30 à 60	pts_moy / 40	Distance maximum pas adaptée											Pas de faces extraites
d_pts * 2	30	pts_moy / 10	1	4	0	2	0	2	2	7	5	15		
d_pts * 2	30	pts_moy / 20	7	2	2	0	0	2	5	4	231	8		
d_pts * 2	30	pts_moy / 30	8	1	3	0	1	1	5	4	22	6		
d_pts * 2	30	pts_moy / 40	9	0	4	0	2	0	9	1	32	1	ok mais un peu trop de faces	
d_pts * 2	40	pts_moy / 10	2	4	0	2	0	2	3	5	6	13		
d_pts * 2	40	pts_moy / 20	5	2	2	0	0	2	5	3	49	7		
d_pts * 2	40	pts_moy / 30	6	1	3	0	2	0	5	3	33	4		
d_pts * 2	40	pts_moy / 40	7	0	3	0	2	0	9	0	33	0	Solution bonne, mais un peu trop de faces	
d_pts * 2	50	pts_moy / 10	2	4	1	1	0	2	3	5	6	12		
d_pts * 2	50	pts_moy / 20	5	0	2	0	0	2	4	2	30	4		
d_pts * 2	50	pts_moy / 30	5	0	3	0	2	0	5	1	34	1	pièce 12 1 face juste un peu grande	
d_pts * 2	50	pts_moy / 40	5	0	3	0	3	0	7	0	14	0	Solution bonne	
d_pts * 2	60	pts_moy / 10	2	3	1	1	0	2	3	6	8	12		
d_pts * 2	60	pts_moy / 20	3	0	2	0	2	0	5	1	16	1	Les faces s'étalent trop	
d_pts * 2	60	pts_moy / 30	3	0	3	0	2	0	6	0	16	0	Les faces s'étalent trop	
d_pts * 2	60	pts_moy / 40	3	0	3	0	2	0	7	0	16	0	Les faces s'étalent trop	
d_pts * 3	30	pts_moy / 10											Nombre de points minimum pas adapté	
d_pts * 3	30	pts_moy / 20	7	1	2	0	0	2	5	3	308	6		
d_pts * 3	30	pts_moy / 30	8	0	3	0	1	1	6	3	30	4		
d_pts * 3	30	pts_moy / 40	8	0	4	0	2	1	9	0	39	1		
d_pts * 3	40	pts_moy / 10											Nombre de points minimum pas adapté	
d_pts * 3	40	pts_moy / 20	5	1	6	0	0	2	5	2	49	5		
d_pts * 3	40	pts_moy / 30	6	1	3	0	2	0	5	2	37	3		
d_pts * 3	40	pts_moy / 40	7	0	3	0	3	0	9	0	18	0	Solution bonne, mais un peu trop de faces	
d_pts * 3	50	pts_moy / 10											Nombre de points minimum pas adapté	
d_pts * 3	50	pts_moy / 20	5	0	2	0	0	2	4	1	14	3		
d_pts * 3	50	pts_moy / 30	5	0	3	0	2	0	5	1	31	1		
d_pts * 3	50	pts_moy / 40	5	0	3	0	3	0	7	0	20	0	Solution bonne	
d_pts * 3	60	pts_moy / 10											Nombre de points minimum pas adapté	
d_pts * 3	60	pts_moy / 20	3	0	2	0	2	0	5	1	23	1	Les faces s'étalent trop	
d_pts * 3	60	pts_moy / 30	3	0	3	0	2	0	5	0	23	0	Les faces s'étalent trop	
d_pts * 3	60	pts_moy / 40	3	0	3	0	2	0	6	0	22	0	Les faces s'étalent trop	

#### **A.4 Vue 3D du résultat du test de découpage en faces**

Le PDF 3D qui suit montre le découpage des 4 faces choisies pour le test de découpage en faces par propagation des normales avec les paramètres retenus pour la suite du travail.

Comme la visualisation des PDF 3D n'est pas possible avec tous les lecteurs PDF, les nuages de points sont également disponibles dans les annexes numériques.



## **A.5 Vue 3D du remontage complet de Lokalalei 2C**

Le PDF 3D qui suit permet de voir le remontage complet de Lokalalei 2C effectué avec le script.

Comme la visualisation des PDF 3D n'est pas possible avec tous les lecteurs PDF, les nuages de points sont également disponibles dans les annexes numériques.



## **A.6 Vue 3D du remontage partiel de MIBR16**

Le PDF 3D qui suit permet de voir le remontage partiel de MIBR16 effectué avec le script.

Comme la visualisation des PDF 3D n'est pas possible avec tous les lecteurs PDF, les nuages de points sont également disponibles dans les annexes numériques.





## Annexe B

# Mode d'emploi

### B.1 Guide utilisateur

Le présent guide a pour but d'accompagner l'utilisateur qui veut essayer par lui-même de faire un remontage sur un jeu de données appariable. La première partie guide l'utilisateur pour les installations nécessaires, la deuxième explique les paramètres importants à modifier et la troisième donne quelques conseils pour réussir.

#### B.1.1 Installation

L'environnement utilisé pour le développement du projet est décrit ci-dessous. Afin d'éviter les problèmes de compatibilité, il est recommandé de recréer le même environnement pour l'utilisation du script. Cependant, il devrait également être possible de l'utiliser avec des versions plus récentes des bibliothèques.

**Environnement :**

- Python 3.9.7
- Numpy 1.22.3
- Open3d 0.15.1
- Scipy 1.8.0
- Opencv-python 4.5.5.64
- Torch 1.11.0 (<https://pytorch.org/>) Choisir les options désirées et exécuter la ligne de commande indiquée.
- matplotlib 3.5.1 (uniquement pour de l'affichage des images et filtres.)

### B.1.2 Utilisation

**Paramètres utiles :** Des paramètres peuvent être réglés pour trois phases du calcul : découpage des faces, création des images et filtrage. Ils sont regroupés au début du script.

#### Découpage des faces

- **mypath :** Donner le chemin d'accès aux données.
- **cloud\_resolution :** Il faut entrer ici la résolution du nuage de point. Elle sera utilisée pour déterminer la distance de recherche maximum des points voisins. Dans le cas où la résolution du nuage n'est pas homogène, prendre plutôt la valeur haute pour avoir des faces partout.
- **cloud\_mean\_pts :** Il faut entrer ici le nombre de points moyen par nuage. Ce paramètre est utilisé pour déterminer la taille minimum que peut avoir une face. Si la taille des nuages est très différente, se baser plutôt sur les petits nuages.
- **neighbour\_normals\_nbr :** En règle générale, il n'est pas nécessaire de modifier ce paramètre. Il donne le nombre de voisin à prendre en compte pour l'orientation des normales des points par rapport au plan local.
- **angle\_max :** Il s'agit de l'angle maximum entre la normale moyenne des points de la face et la normale du point à tester. Si l'angle est inférieur à cette limite, le point à tester est inclus dans la face, sinon il est rejeté. Se référer au Chapitre 3.2.1 pour l'influence du paramètre.

#### Création des images

- **pixel\_size :** Il faut entrer ici la résolution du nuage de point. La taille du pixel sera utilisée pour déterminer la taille des images. Dans le cas où la résolution du nuage n'est pas homogène, prendre plutôt la valeur basse pour pouvoir apparier les zones qui ont plus de relief. Se référer au Chapitre 3.2.2 pour l'influence de la taille de l'image.
- **radius\_value :** Le rayon en pixel sera utilisé pour le calcul du VO. Sur les tests menés, il ressort que le VO de 5 permet généralement les meilleurs appariements. Si le rayon est modifié, il faut également modifier en conséquence "pixel\_size".

#### Filtrage

- **inliers\_min :** Il faut entrer ici le nombre minimum de correspondances qu'il faut pour qu'un appariement soit considéré comme correct dans le premier filtre. Se référer au chapitre 3.2.3 pour plus d'informations.
- **iterations\_max :** Il faut entrer ici le nombre maximum d'itérations que le calcul de moindres carrés peut faire pour qu'un appariement soit considéré comme correct dans le deuxième filtre.

### B.1.3 Conseil

L'appariement est entièrement basé sur la géométrie des objets. Les objets érodés ou déformés ne pourront pas être appariés. L'acquisition numérique doit refléter la géométrie de l'objet, le bruit d'acquisition peut avoir une grande incidence sur le calcul du VO. Plus la matière est lisse et moins il sera possible de sortir des appariements entre les pièces.

## B.2 Guide programmeur

Le présent guide a pour but d'informer l'utilisateur qui veut essayer par lui-même de compléter ou améliorer le code. La première partie détaille les fonctions existantes et la deuxième propose quelques pistes d'amélioration. Les aspects relatifs à l'installation et à l'environnement utilisé sont décrits dans l'Annexe B.1 et ne sont pas répétés ici.

### B.2.1 Description des fonctions

- **import\_data(mypath)** : La fonction importe les données à partir du chemin spécifié. Elle retourne une liste de nuage sous forme de nuages de points Open3d et une liste de noms de fichiers.
- **propage\_normal(pcd, id\_pt\_query, d\_max, dot\_min, pcd\_tree)** : La fonction permet d'extraire une face par propagation de normales à partir du nuage de points, d'un point de départ, d'un critère de distance et d'une valeur minimum de cosinus qui découle de l'angle maximum réglé en paramètre. Elle retourne les ID des points de la face et la normale moyenne des points de la face.
- **extra\_inliers(pcd, id\_pt\_query, d\_max, pcd\_tree)** : Même fonction que la précédente sans limite d'angle. Elle permet d'appliquer un "buffer" aux points de la face et ainsi d'inclure les points proches qui ne satisfaisaient pas aux critères pour éviter les petits trous.
- **propage\_normal\_all(pcd, d\_max, angle\_max, nb\_pts\_min)** : La fonction utilise les deux précédentes sur l'ensemble d'un nuage jusqu'à ce que toutes les faces supérieures au nombre de point minimum paramétré soit extraites. Retourne une liste de faces et normales sous forme de matrice pour le nuage traité.
- **orient\_plan(normal\_plan, inlier\_pcd)** : La fonction oriente la face de manière à ce que sa normale soit verticale. Il s'agit d'une préparation pour que l'image soit perpendiculaire à la face. Retourne la face orientée sous forme de matrice et la rotation sous forme de rotation Scipy.
- **max\_size(plans\_list)** : La fonction détermine la plus grande emprise en X ou Y pour que cette distance soit utilisée comme référence à la création des images. Retourne la taille en unité réelle.
- **size\_DEM = round(xy\_max\_size / pixel\_size)** : Détermination de la taille de l'image par rapport à la plus grande face et en fonction de la taille de pixel choisi.
- **init\_SIFT()** : Si le calcul est lancé sur une machine sans GPU, il est possible de modifier le paramètre dans cette fonction. Par défaut, le calcul se fait sur le GPU. Il est également possible de modifier le nombre maximum de points d'intérêts qui peuvent être détectés par image dans cette fonction. Le nombre est fixé à 3000 par défaut.
- **prepare\_VO(radius\_value)** : La fonction prépare les constantes relatives au rayon paramétré en prévision du calcul du VO. Repris de [Rolland et al., 2022].
- **DEM(array\_pcd)** : La fonction crée une grille régulière de la taille d'image déterminée plus haut. Les valeurs Z sont insérées dans la grille par interpolation linéaire. D'autres méthodes d'interpolation peuvent être utilisées, mais elles n'ont pas été testées dans le cadre de ce travail. Retourne la grille régulière et les valeurs minimum en x et en y en prévision du retour aux coordonnées du nuage.
- **array\_image(image\_input, p1, p2)** : La fonction élimine les valeurs extrêmes de manière à améliorer le contraste de l'image. Elle est appliquée sur les images en sortie du VO. Les paramètres p1 et p2 ont été déterminés arbitrairement sans procéder à des tests. Ils peuvent faire l'objet de tests permettant de voir leur influence. Retourne l'image contrastée.
- **calculations\_VO(grid\_z)** : Calcul du VO selon [Rolland et al., 2022]. Les options de calcul VOP et VON ne sont pas implémentées ici. L'option de floutage n'a pas non plus été récupérée, il est possible qu'elle puisse être utile pour enlever le bruit des nuages si nécessaire. De plus, pour les besoins de l'appariement SIFT qui suivra, une image inversée selon Figure 2.8 est

- également créée. Retourne l'image et l'image inversée.
- **det.detectAndCompute(img, None)** : Extrait une liste de points d'intérêt et les descripteurs associés.
  - **Stratégie d'appariement** : Les fonctions pour l'appariement sont reprises de Dmytro Mishkin, "[https://local-features-tutorial.github.io/pdfs/Local\\_features\\_from\\_paper\\_to\\_practice.pdf](https://local-features-tutorial.github.io/pdfs/Local_features_from_paper_to_practice.pdf)" pour les détails. Il en ressort l'homographie entre les deux images, le masque indiquant les bonnes et mauvaises correspondances de l'homographie, les coordonnées sources et de destination des correspondances dans les images et le nombre de correspondances. L'homographie n'est pas utilisée par la suite, elle sert juste à indiquer que l'appariement n'est pas possible en cas d'homographie non trouvée.
  - **coord\_img\_2\_cloud(mask, src, dst)** : La fonction retransforme les coordonnées images en coordonnées 3D. Seuls les points comptés comme bonnes correspondances sont retransformés. Retourne des listes de coordonnées source et destination entre les faces appariées.
  - **selection\_1(Inliers)** : La fonction sélectionne les solutions qui ont un nombre de correspondances supérieur à la limite paramétrée. Retourne une liste de solutions ayant passé le test.
  - **transform(D, trans)** : La fonction transforme une liste de points 3D en fonction des paramètres Omega, Phi, Kappa, tx, ty et tz (3 rotations et 3 translations). Retourne les coordonnées transformées.
  - **projection(O, D, trans)** : La fonction calcule la différences de coordonnées dX, dY et dZ entre deux listes de points appariées. Retourne les dX, dY et dZ regroupés dans un tableau.
  - **MS(O, D, trans\_init, max\_step = 50, epsilon = 0.001)** : La fonction calcul Omega, Phi, Kappa, tx, ty et tz en fonction des coordonnées 3D des correspondances trouvées dans l'homographie. Un nombre maximum d'itérations et un accroissement minimal sont mis en place comme critères d'arrêt. Retourne les 6 paramètres de transformation et le nombre d'itérations.
  - **wpk\_to\_euler(omega, phi, kappa)** : Transformation de convention d'écriture de rotation entre Omega, Phi, Kappa et Euler. Retourne rotation selon convention d'Euler.
  - **regroup\_transformations(O\_cloud, O\_face, D\_cloud, D\_face, trans)** : En fonction des nuages et faces source et de destination, regroupe les trois transformations nécessaires pour passer du nuage source au nuage de destination. Retourne nuage source, nuage de destination et transformation.
  - **invert\_transformation(T)** : La fonction renvoie la transformation destination => source à la place de source => destination.
  - **global\_transformation(Trans, Cloud)** : La fonction transforme les transformations locales en transformations globales et les applique aux nuages de points. Retourne les nuages transformés et les n° des nuages assemblés.

### B.2.2 Pistes d'améliorations

**Extraction des faces** : Plusieurs améliorations peuvent encore être apportées à l'extraction des faces. Dans certains cas, des trous sont visibles à l'intérieur des faces. Pour certains, ils sont justifiés car l'inclusion des points créerait des surplombs dans la face et ne permettrait pas de faire une image propre par la suite, mais certains pourraient encore être évités car l'angle maximum les a exclus. Un deuxième tri à l'intérieur du périmètre de la face pour les points non inclus pourrait être fait sur l'idée de la fonction "extra\_inliers" mais avec une distance de récupération plus grande et uniquement à l'intérieur du périmètre de la face. L'autre possibilité d'amélioration pour l'extraction des faces serait de bloquer le plan et sa normale après un certain nombre de points inclus pour ne pas faire de faces trop grandes dans les pièces à courbure faible sans arête.

**Calcul du VO** : Seule la partie du calcul de VO a actuellement été reprise. L'option de floutage n'a pas été récupérée pour ne pas dégrader la qualité des données, mais dans le cas de données légèrement bruitées, il pourrait être utile d'ajouter cette option.

**Calcul des moindres carrés :** Bien qu'une vérification géométrique des correspondances ait déjà été faite lors de l'homographie, il est possible que certaines correspondances ne soient pas correctes en 3D. Le calcul des moindres carrés pourrait encore être couplé à un RANSAC afin de s'assurer qu'il n'y ait plus de mauvaises correspondances. De plus, une seule valeur de stop est prévue pour le moment. Il serait possible d'en prévoir deux car les angles et les distances ont des ordres de grandeur de précision différents.

**Transformation globale :** Pour le moment, la fonction de transformation globale ne fait que d'utiliser les transformations locales les unes après les autres. La possible surabondance de transformations n'est pas exploitée. Il serait possible de mettre en place un contrôle de cohérence entre les transformations dans le but de repérer si une mauvaise transformation est encore présente dans les solutions. D'autre part, une optimisation globale des transformations en utilisant également les liens qui n'ont pas été détectés dans un premier temps mais qui sont repérables après le calage grossier global.

**Itérations :** Une fois le premier remontage effectué, de nouvelles faces peuvent apparaître sur l'objet remonté. Un redécoupage en faces de l'objet remonté couplé à une deuxième itération du calcul sur les pièces non appariées peuvent conduire à la découverte d'appariements supplémentaires.

**Temps de calcul :** Pour gagner du temps de calcul, deux axes peuvent être développés. Le premier concerne les opérations initiales pour la création des faces et images ainsi que la détection des points d'intérêts. Chaque opération est faite sur tous les nuages à la suite. Comme il n'y a pas d'interaction entre les différents nuages à ce stade, il serait possible de paralléliser ces opérations de calcul qui, de surcroît, sont celles qui prennent le plus de temps dans le calcul. Le deuxième axe concerne le langage de programmation. Le prototype a été codé en Python car il s'agit d'un langage pratique pour effectuer différents tests et recherches. Néanmoins, pour l'utilisation avec un grand nombre de données, il serait préférable de coder en C++ pour gagner du temps de calcul.



# Table des figures

1.1	Différentes méthodes de remontage de poterie. . . . .	3
1.2	Appariement en imagerie médicale. . . . .	4
1.3	Image d'illustration : Remontage lithique. <i>Source : www.paleotime.fr</i> . . . . .	4
1.4	Huit cadres d'animation qui motivent le nom spin-image. Le processus de génération de spin-image peut être visualisé comme une feuille tournant autour de la normale du point orientée, accumulant des points comme s'il balaye l'espace [Johnson, 1997].	5
1.5	Appariement de deux nuages de points à l'aide de l'ISS. (a) nuage de points source de la vue partielle d'une voiture en bleu et le nuage de point de modèle en jaune ; (b) points de base saillants extraits marqués en rouge ; (c) appariement de paires de descripteurs entre la requête et le modèle ; (d) estimation de rotation à partir de paires de descripteurs correspondants : chaque point correspond à la rotation « observation » d'une paire d'appariements de descripteurs de forme intrinsèques, avec la couleur indiquant la densité de l'estimation (vert pour la faible densité et rouge pour la densité élevée) ; (e) appariement de paires de descripteurs correspondant au plus grand groupe de rotations ; f) alignement du nuage source sur le modèle en utilisant la rotation estimée [Zhong, 2009]. . . . .	6
1.6	Schéma de construction des descripteurs des catégories de signatures et d'histogrammes [Salti et al., 2014]. . . . .	7
1.7	Carte de profondeur d'une face de fragment rocheux . . . . .	8
1.8	Détermination du Sky view factor pour un point donné [Yi and Kim, 2017]. . . . .	9
1.9	Détermination du VO, VO <sub>P</sub> , VON pour un point donné [Rolland et al., 2022]. . . . .	9
1.10	Comparaison entre VO, VO <sub>P</sub> , VON et SVF pour une application dans différents cas donnés [Rolland et al., 2022]. . . . .	10
2.1	La chaîne de traitement utilisée pour le remontage des objets. . . . .	13
2.2	Jeu de données de deux blocs-sièges utilisé pour illustrer la méthodologie. . . . .	14
2.3	Création et orientation des normales pour un nuage de points. . . . .	15
2.4	Coupe de principe d'extraction des plans avec la méthode de détection du meilleur plan par 3 points. . . . .	16
2.5	Comparaison de l'extraction des faces selon les deux méthodes présentées. . . . .	17
2.6	Carte de profondeur d'une face appariale de la pièce test. . . . .	18
2.7	VO calculé sur la base de la carte de profondeur de la Figure 2.6. . . . .	19
2.8	Création des images VO pour permettre l'appariement des images. . . . .	19
2.9	Appariement de deux faces avec en jaune les correspondance trouvées entre les points d'intérêts des deux images. . . . .	20
2.10	Transformation pour amener le nuage source vers le nuage de destination. . . . .	21
2.11	Inspection 3D nuage à maillage entre le nuage source transformé et le nuage destination.	21
2.12	Détection des points d'intérêt du nuage. . . . .	22
2.13	Calage grossier à l'aide des points d'intérêt. . . . .	22
2.14	Calage grossier de faces à l'aide des points d'intérêt. . . . .	23
2.15	Calage fin ICP. . . . .	24

2.16	Résultat de filtrage des solutions d'appariement entre chaque face du bloc 1 et chaque face des deux blocs-sièges de l'Annexe A.2, soit 5 faces et 6 faces donc 30 solutions possibles. . . . .	25
2.17	Résultat de filtrage des solutions restantes après le premier tri. La solution sous le seuil est correcte, les autres sont fausses. . . . .	26
2.18	Résultat de filtrage des solutions sur le nombre de correspondances. . . . .	27
2.19	Résultat de filtrage des solutions sur la zone de chevauchement. . . . .	27
2.20	Résultat de filtrage des solutions sur l'erreur moyenne quadratique des correspondances. . . . .	28
3.1	Les 13 pièces du remontage Lokalalei 2C étalées. . . . .	31
3.2	Les 16 pièces du remontage MIBR16 étalées. . . . .	32
3.3	Les deux faces du jeu JGBO1 utilisées pour les tests. . . . .	32
3.4	Les quatre pièces du Lokalalei 2C utilisées pour le test de découpage. . . . .	33
3.5	Influence des paramètres de rayon du VO et taille d'image sur le nombre de correspondances trouvées pour le jeu de données Lokalalei 2C. . . . .	35
3.6	Influence des paramètres de rayon du VO et taille d'image sur le nombre de correspondances trouvées pour le jeu de données MIBR16. . . . .	35
3.7	Visualisation de la différence de résolution sur les pièces du jeu MIBR16. . . . .	36
3.8	Les solutions de la section verte sont correctes, celles de la section jaune doivent être validées par un deuxième filtrage, celles de la section rouge n'ont pas été étudiées car les chances d'y trouver des solutions correctes sont très faibles. . . . .	37
3.9	Nombre de solutions justes et fausses pour un seuil fixé à 10 correspondances entre images. . . . .	38
3.10	Nombre de solutions justes et fausses pour un seuil fixé à 15 correspondances entre images. . . . .	38
3.11	L'appariement entre les deux faces a été trouvé 24 fois pour le nuage à pleine résolution. . . . .	39
3.12	L'appariement entre les deux faces a été trouvé 13 fois pour le nuage à une demi résolution. . . . .	39
3.13	L'appariement entre les deux faces a été trouvé 4 fois pour le nuage à un quart de résolution. . . . .	39
3.14	Remontage du Lokalalei 2C. . . . .	40
3.15	Statistiques sur le traitement des jeux de données Lokalalei 2C et MIBR16. . . . .	40
3.16	Remontage du MIBR16. . . . .	41
3.17	Pièces restantes après le remontage. . . . .	41
3.18	Correspondances trouvées pour l'appariement entre deux pièces du MIBR16. . . . .	42
3.19	Les différents blocs remontés automatiquement par le script. . . . .	43
3.20	Zone utilisable pour l'appariement après traitement de l'image par le VO. . . . .	44
3.21	Répartition des points d'intérêts en fonction de la rugosité. . . . .	45



# Bibliographie

- [Barreau et al., 2014] Barreau, J.-B., Nicolas, T., Bruniaux, G., Petit, E., Petit, Q., Bernard, Y., Gagne, R., and Gouranton, V. (2014). Ceramics fragments digitization by photogrammetry, reconstructions and applications. *1412.1330*.
- [Besl and McKay, 1992] Besl, P. J. and McKay, N. D. (1992). Method for registration of 3-D shapes. In Schenker, P. S., editor, *Sensor Fusion IV : Control Paradigms and Data Structures*, volume 1611, pages 586 – 606. International Society for Optics and Photonics, SPIE.
- [Bradski, 2000] Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.
- [Delagnes and Roche, 2005] Delagnes, A. and Roche, H. (2005). Late pliocene hominid knapping skills : The case of lokalalei 2c, west turkana, kenya. *Journal of Human Evolution*, volume 48 :page 435–472.
- [Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, volume 24 :page 381–395.
- [hui Liao et al., 2020] hui Liao, S., Xiong, C., Liu, S., qi Zhang, Y., and lin Peng, C. (2020). 3d object reassembly using region-pair-relation and balanced cluster tree. *Computer Methods and Programs in Biomedicine*, volume 197 :page 105756.
- [Johnson, 1997] Johnson, A. E. (1997). Spin-images : a representation for 3-d surface matching.
- [Kampel and Sablatnig, 2004] Kampel, M. and Sablatnig, R. (2004). 3d puzzling of archeological fragments. In *Proc. of 9th Computer Vision Winter Workshop*, volume 2, pages 31–40. Slovenian Pattern Recognition Society.
- [Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, volume 60 :page 91–110.
- [Rolland et al., 2022] Rolland, T., Monna, F., Buoncristiani, J. F., Magail, J., Esin, Y., Bohard, B., and Chateau-Smith, C. (2022). Volumetric obscurance as a new tool to better visualize relief from digital elevation models. *Remote Sensing*, volume 14 :page 941.
- [Salti et al., 2014] Salti, S., Tombari, F., and Di Stefano, L. (2014). Shot : Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, volume 125 :pages 251–264.
- [Waldhäusl and Ogleby, 1994] Waldhäusl, P. and Ogleby, C. (1994). 3 x 3 rules for simple photogrammetric documentation of architecture. *International Archives of Photogrammetry and Remote Sensing*, volume 30 :pages 426–429.
- [Yi and Kim, 2017] Yi, Y. K. and Kim, H. (2017). Universal visible sky factor : A method for calculating the three-dimensional visible sky ratio. *Building and Environment*, volume 123 :pages 390–403.
- [Zhong, 2009] Zhong, Y. (2009). Intrinsic shape signatures : A shape descriptor for 3d object recognition. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 689–696.

- [Zhou et al., 2018] Zhou, Q.-Y., Park, J., and Koltun, V. (2018). Open3D : A modern library for 3D data processing. *1801.09847*.
- [Üçoluk and Hakkı Toroslu, 1999] Üçoluk, G. and Hakkı Toroslu, I. (1999). Automatic reconstruction of broken 3-d surface objects. *Computers & Graphics*, volume 23 ;pages 573–582.