



HAL
open science

Extraction de paramètres bio-geo-physiques de surfaces 3D reconstruites par multi-stéréo-restitution d'images prises sans contraintes

Benoit Petitpas

► **To cite this version:**

Benoit Petitpas. Extraction de paramètres bio-geo-physiques de surfaces 3D reconstruites par multi-stéréo-restitution d'images prises sans contraintes. Autre. Université Paris-Est, 2011. Français. NNT : 2011PEST1042 . tel-00693326

HAL Id: tel-00693326

<https://theses.hal.science/tel-00693326>

Submitted on 2 May 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ PARIS-EST
ÉCOLE DOCTORALE MSTIC

Thèse de doctorat

Science de l'Information Géographique

BENOIT PETITPAS

**EXTRACTION DE PARAMÈTRES BIO-GEO-PHYSIQUES DE SURFACES 3D
RECONSTRUITES PAR MULTI-STÉRÉO-RESTITUTION D'IMAGES PRISES
SANS CONTRAINTES**

Thèse dirigée par Jean-Paul Rudant, Michel Roux et Laurent Beaudoin.

Soutenue le 15 décembre 2011.

Jury :

Marc Pierrot-Deseilligny

Pierre Grussenmeyer

Marc Dubois

Sébastien Valette

Bernard Riera

Rapporteur

Examineur

*Appelez-moi Ismaël. Il y quelques années de cela — peu importe combien exactement
— comme j'avais la bourse vide, ou presque, et que rien d'intéressant ne me retenait
à terre, l'idée me vint de naviguer un peu et de revoir le monde marin.*

Moby-Dick,
Herman Melville,
Trad. Philippe Jaworski, collection de la Pléiade, 2006.

*Call me Ishmael. Some years ago — never mind how long precisely — having little or
no money in my purse, and nothing particular to interest me on shore, I thought I
would sail about a little and see the watery part of the world.*

Moby-Dick,
Herman Melville, 1851

remerciements

Je souhaite tout particulièrement remercier mes directeurs de thèse qui m'ont été d'une aide exemplaire tout au long de cette thèse. En premier Jean-Paul Rudant pour avoir échafaudé le projet et le sujet de cette thèse et qui m'a accompagné au travers de réunions impromptues et de débats énergiques sur l'orientation de cette thèse. Je souhaite tout particulièrement remercier la personne sans laquelle cette thèse n'aurait pas existé, Michel Roux qui fut un modèle de rigueur pour moi, sa gentillesse n'a d'égale que son courage à corriger mes fautes d'orthographe trop nombreuses. Son engagement constant dans cette thèse m'a toujours permis de trouver une issue aux problèmes qui me bloquaient. Enfin je souhaite remercier Laurent Beaudouin qui malgré le peu de temps passé ensemble a sut à chaque fois m'insuffler son enthousiasme débordant. Venir à l'esiea manger un bo-bun, en compagnie d'Antoine Gademer et du reste de l'équipe ATIS, en conversant sur ma thèse dans une bonne humeur communicative a toujours réussi à m'ammener de nouvelles idées.

Je souhaite remercier le jury d'avoir bien voulu s'associer à cette thèse, en commençant avec le président de jury M. Marc Dubois, chercheur au CEA. Je remercie les rapporteurs de cette thèse, M. Pierre Grussenmeyer, Professeur des Universités, Département Génie Civil -Topographie de l'insa Strasbourg et responsable du laboratoire de Photogrammétrie et M. Marc Pierrot-Deseilligny, directeur de recherche au laboratoire MATIS. Je remercie de même les examinateurs d'avoir accepté mon invitation, M. Bernard Riera chargé de recherche au CNRS au muséum national d'histoire naturelle, Sébastien Valette, chargé de recherche CNRS à l'INSA Lyon.

L'amitié est chose peut commune et je me targue d'avoir les meilleurs du monde et nos virés dans Paris ont toujours sut me rappeler que parler travail quand on est plus au travail est un péché grave. Merci à vous Romuald, Nicolas, Morrigan, Etienne, Guillaume, Benny ...

Que serait une thèse sans les compagnons d'aventure que sont nos collègues thésards. La présence à mes cotés de mon acolyte et ami Charles-Alban fut pour moi toujours source de bonheur et de travail bien fait. Mieux que google, il est ma source intarissable de réponses aux questions insolubles de programmation que cette thèses a mis sur ma route. Thibault Napoléon reparti trop tôt pour sa Normandie natale. Je regrette déjà tous les compagnons de la C32 qui sont passé mais qui ne restèrent pas assez longtemps, Aymen Shabou, Dina Hadidi, Guillaume Hochard, Talita Perciano, Hélène Sportouche et Flora Dellinger. Je remercie aussi le bureau cool, alors pour ne pas tous les cité : Vincent Duval, Ivan

Kyrgyzof, Marie Lienou pour les plus anciens membres, Gui-song Xia, Cécilia Aguerbere, Batiste Mazain, Guillaume Tartavelle, Alasdair Newson, Yann traonmilin pour les plus récents et tous les autres que j'oublie sûrement. J'en profite pour remercier Adrian Marquès de sa gentillesse proverbiale et d'avoir fait un si bon ami aussi cours que fut sa présence et un fan, comme moi, de Terry Pratchett. Je fait un salut spécial au second Stéréo-Vionneur du LTCI, Thierry Guillemot dont les discussions m'ont permis d'avancer très rapidement sur le lissage 3D, sa bibliographie confrontée à la mienne nous a permis d'avoir de nombreuses conversations instructives. Notre collaboration fut fructueuse et notre amitié encore plus.

Enfin citer les collègues sans parler des réunions hebdomadaires à la taverne de la butte aux caillles serait une insulte au groupe 3D (CGgroup), groupe mené par le duo Tamy Boubekour, qui fut d'une grande aide et que je remercie pour toutes les discussions que nous avons eu, et Elmarr Eisemann, l'homme le plus gentil de l'histoire. Ce duo se complète de Julien Tierny le ch'ti reconverti en Vincennois reconverti en Parisien et Pooran Memari. Enfin les thésards : Jean-Marc Tierny, Noura Faraj, Mathias Höllander, Guillaume Vialanex (que je remercie pour m'avoir fait découvrir les jeux de société, même s'il n'aime pas smallworld), et le post-doc Tobias Rietchel reparti en Allemagne et qui n'a toujours pas compris mon humour.

Enfin à mes parents adorés, pour leurs patiences, pour leurs gentilleses et surtout pour les corrections orthographiques.

Cette thèse n'est rien sans la présence à mes cotés de ma femme, Anne-Claire qui aura sut jour après jour trouver les mots pour me soutenir quand tout allait mal et rire quand tout allait bien. Elle est tout pour moi, je lui dédie ce travail.

résumé

Extraire des mesures sur des surfaces est un problème dans de nombreux domaines de recherche. L'archaïsme de certains systèmes ou la cherté d'appareils perfectionnés empêchent l'extraction rapide et robuste de ces paramètres. Pourtant, ils sont essentiels dans de nombreux domaines, comme les paramètres de rugosité qui interviennent dans de nombreux phénomènes physiques ou les valeurs dendrométriques pour l'étude de la bio-diversité. En parallèle, l'utilisation et la production de contenu 3D se développent considérablement ces dernières années dans des domaines très divers.

Le but de cette thèse est donc d'utiliser toutes ces innovations dans le domaine de la reconstruction 3D et de les appliquer à la mesure de paramètres de surface. Pour cela, il est nécessaire de créer une chaîne complète de reconstruction 3D, n'utilisant que des images prises sans contrainte dans le but d'être accessible au plus grand nombre.

Dans cette chaîne nous utilisons des algorithmes de stéréo-vision robustes produisant, pour chaque couple d'images, un nuage de points. Après le passage de ces nuages dans un référentiel commun, une étape de filtrage des points 3D et de suppression des redondances est nécessaire. Une étape de lissage permet d'obtenir le nuage final. Pour justifier des bons résultats obtenus, une étape de validation nous a permis de vérifier et d'étudier la robustesse de la chaîne de traitements développée. Enfin, les paramètres de rugosité et dendrométriques seront extraits. Nous étudierons dans ces deux cas, comment extraire ces informations et leurs utilisations.

abstract

Extracting measures on surfaces is a problem in many areas of research. The archaism of some systems or the costliness of sophisticated devices prevent the fast and robust extraction of these parameters. Yet these measures are essential in many areas, such as roughness parameters involved in many physical phenomena or dendrometric values for the study of biodiversity. In parallel, the use and production of 3D content has grown dramatically this past years in very diverse domains.

The purpose of this thesis is to use these innovations in the context of surfaces parameter measurements. It is necessary to create a complete chain of 3D reconstruction, using pictures taken without constraint, in order to be open to as many people.

This chain uses robust stereo-vision algorithms in order to produce a point cloud for each pair of images. After the generation of these point cloud in the same geometric frame, a filtering step of 3D points and a deletion step of redundancies are necessary and a smoothing step allows us to obtain the final point cloud. To reveal the good results, a validation step has enabled us to verify and investigate the robustness of the developed chain. The roughness and dendrometric parameters are finally extracted. We will study in both cases, how to extract this information and their uses.

Table des matières

1	Reconstruction 3D	13
1.1	Données	14
1.1.1	Modèle projectif	14
1.1.2	Modèle stéréoscopique	16
1.1.3	Précision des mesures de profondeurs souhaitées	17
1.2	Appariement de points d'intérêt	18
1.2.1	Détections	18
1.2.2	Descripteurs	19
1.2.3	Appariements	19
1.2.4	Résultats et discussions	20
1.3	Géométrie épipolaire	22
1.3.1	Rectification épipolaire	24
1.4	Appariement dense	28
1.4.1	Méthodes globales	34
1.4.2	Approche locale multi-échelles	35
1.5	Auto-calibrage	42
1.5.1	Auto-calibrage d'un couple d'image	44
1.5.2	Ajustement de faisceaux	47
1.6	Triangulation et résultats	49
1.6.1	Résultats et discussions	50
2	Traitement 3D	56
2.1	Suppression des outliers par analyse en composante connexe	58
2.1.1	Calcul des k-plus proches voisins	58
2.1.2	Analyse en composantes connexes	61
2.2	Suppression des points redondants	67
2.2.1	Recalage des nuages	67
2.2.2	Suppression des redondances par filtrage colorimétrique	67
2.3	Lissage du nuage de points par moindres carrés glissants	72
2.3.1	Définition générale des MLS	74
2.3.2	État de l'art	74
2.3.3	Surface MLS de projections	76
2.3.4	Surface de projection colorimétrique	79
2.3.5	Résultats et comparaisons	82

3	Résultat, validation et discussion	89
3.1	Résultats	90
3.1.1	Bâtiments	90
3.1.2	Objets	90
3.1.3	Visages	92
3.1.4	Comparaisons	92
3.2	Validation	96
3.2.1	Banc de test	98
3.2.2	De l'espace métrique à l'espace euclidien	99
3.2.3	Résultats des tests	99
3.3	Discussions	103
3.3.1	Étude de l'erreur de disparité	103
3.3.2	Étude de l'erreur de calibrage	105
3.3.3	Conseil de prises de vues	110
4	Mesures de paramètres de surfaces	115
4.1	Paramètres de rugosité	115
4.1.1	Calcul de la rugosité	117
4.1.2	Utilité de la rugosité	119
4.1.3	Discussions et perspectives	120
4.2	Paramètres dendrométriques	121
4.2.1	Paramètres de longueurs	121
4.2.2	Paramètres d'épaisseur	123
4.3	Perspectives	124
4.3.1	Utilisation d'un quadri-coptère	124
4.3.2	Perspectives	139
4.3.3	MobyDick	140

Introduction

Contexte de la thèse

Il est intéressant de mettre en rapport l'évolution des techniques de reconstruction 3D, qui ces dernières années ont offert des résultats de plus en plus robustes, avec l'utilisation dans différents domaines de recherche de matériels de mesures anciens et parfois peu adaptés. En particulier dans deux domaines où leur utilisation est longue et difficile à mettre en place, l'extraction de paramètres de rugosités et les paramètres dendrométriques, définissant l'ensemble des paramètres caractérisant un individu végétal.

Cette thèse a donc pour but de palier aux problèmes de mesures de paramètres sur des surfaces au moyen de techniques modernes de reconstruction 3D. Pour cela nous mettrons au point une chaîne complète de reconstruction 3D, de filtrage 3D et de mesures automatiques pour permettre l'extraction des paramètres de rugosité ou de dendrométrie.

La rugosité est un paramètre très important dans l'étude de nombreux domaines de recherches que nous tentons de lister :

- l'étude de l'érosion [van Donk and Skidmore, 2003], [Zobeck and Onstad, 1987],
- l'hydrologie [Baghdadi et al., 2008], [Verhoest et al., 2008], [Engman and Chauhan, 1995],
- l'agriculture [Davidson et al., 2000], [van Donk and Skidmore, 2003], [Baghdadi et al., 2008].
- l'humidité du sol par étude de la rétrodiffusion radar [Ulaby et al., 1979], [Zribi et al., 2000],

Or, la manière de mesurer cette rugosité reste, dans l'ensemble des cas, compliquée avec les moyens disponibles. Il existe deux types de rugosimètres (montrés dans la figure 1) :

- le rugosimètre à aiguille. Il s'agit d'un cadre en bois, long d'un mètre généralement, sur lequel des aiguilles métalliques amovibles sont implantées. Une fois ce cadre mis à niveau sur le sol, les aiguilles sont mises en contact avec la surface à relever. Une photo est prise du profil donné par les aiguilles. Une fois les photos mises bout à bout, un profil est extrait. Les défauts de ce type d'appareil sont :
 - la fiabilité des résultats ;
 - le temps et la difficulté de mise en place ;
 - le manque de précision ;

- le rugosimètre laser. Il s’agit d’un laser placé sur une tige de fer et capable de faire des aller-retours. Une fois le dispositif mis à niveau, le laser prend un profil. Ces profils pour être exploités nécessitent un calibrage très précis de l’appareil. Les défauts de ce type d’appareil sont :
 - le coût financier ;
 - le temps et la difficulté de mise en place ;
 - le calibrage ;

De plus ces deux appareils ne prennent que des profils, ce qui ne reflète pas parfaitement la réalité du terrain. Enfin ces appareils sont invasifs pour les surfaces étudiées et ont tendance à modifier les valeurs de rugosité par leur passage, même si ce dernier défaut est plutôt l’apanage du rugosimètre à aiguilles.

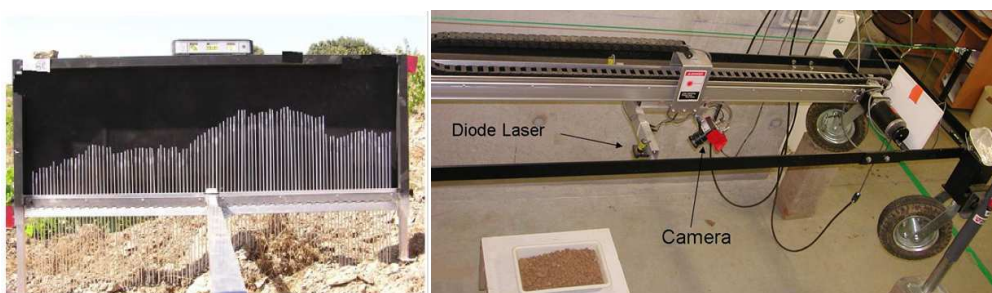


FIGURE 1 – Rugosimètres, aussi appelés profilomètres

Nous tenterons donc d’apporter une nouvelle méthodologie de mesures sur une surface et non plus sur des profils. Pour cela, nous modifierons la définition classique des paramètres de rugosité sur des profils pour qu’ils s’adaptent à des surfaces.

Le deuxième domaine étudié est dans le calcul des paramètres dendrométriques qui sont essentiels à toute étude de la bio-diversité. En effet, on trouve l’intérêt des paramètres dendrométriques dans la compréhension des dynamiques de peuplement servant à l’exploitation forestière ou à la conservation de l’environnement par exemple.

Correspondant aux paramètres caractérisant la végétation, ils sont de natures diverses et, aujourd’hui encore, leurs mesures nécessitent des appareils compliqués et coûteux, aussi bien en temps qu’en matériel. Même si des appareils perfectionnés ont vu le jour pour la mesure de ces paramètres, notamment les hypsomètres capables de mesurer la taille d’un arbre, leurs coûts restent un frein à leur utilisation et de nombreux spécialistes se tournent encore vers les moyens classiques :

- la hauteur d’un arbre est mesurée avec une croix de bûcheron, même si ce système est très ancien et peu précis. Application directe du théorème de Thalès, ce système nécessite de pouvoir mesurer très précisément une longueur au sol, de l’arbre à la personne mesurant, correspondant à la hauteur de l’arbre ;
- la circonférence d’un tronc est mesurée avec un mètre ruban, le diamètre avec un pied à coulisse. Ces techniques, bien qu’assez précises, prennent beaucoup de temps ;
- le dénombrement des espèces s’effectue visuellement. Pour chaque mètre carré

de végétation, une personne va dénombrer les espèces présentes. Cette manipulation nécessite un nombre important de botanistes pour beaucoup de temps ; Tous ces problèmes prouvent à quel point, il est important de trouver une nouvelle méthodologie de mesure.

Dans le même temps la reconstruction 3D est devenue de plus en plus précise et de plus en plus indépendante des contraintes de capteurs. La qualité des reconstructions nous permet aujourd'hui de voir un appareil photo comme un appareil de mesure à part entière.

Cette thèse a donc pour but de fournir aux utilisateurs de ces données un nouveaux moyens d'extraire ces paramètres à travers la reconstruction d'un nuage de point 3D. L'extraction de ces paramètres va nous imposer un certain nombre de contraintes pour que le résultat de ce travail soit utilisé par ceux qui en ont besoin :

- un coût d'extraction faible ;
- un matériel simple et transportable facilement ;
- une automatisation poussée de la reconstruction ;
- une grande robustesse aux scènes déformables, notamment pour l'extraction de paramètres sur la végétation ;

Dans le cadre de ces mesures, la création d'un outil polyvalent de reconstruction 3D générique, MobyDick, a été créé. Il implémente tous les algorithmes nécessaires à l'extraction de paramètres de surfaces de points 3D à partir de photos.

État de l'art de la production de nuage de points 3D

Capter la réalité n'est pas chose aisée. Il y a deux siècles la photographie était encore inconnue. Il y a 20 ans, les techniques photographiques utilisaient des pellicules photos. L'utilisation à l'heure actuelle d'appareils photos numériques, qui utilisent des capteurs CCD, nous permettent la prise d'un grand nombre de photos de bonnes qualités.

Ces photos sont des projections de la réalité dans un plan. Tout le problème de reconstruction 3D réside dans le passage de cette espace plan à l'espace 3D. De nombreuses recherches ont été effectués pour permettre à partir d'images de reconstruire la scène prise en photos.

Pour comprendre l'objet des recherches, il est intéressant de faire un parallèle entre la vision humaine et la photo. En effet, l'être humain est capable de voir en 3D, car il prend, en temps réelle, une photo du monde tous les quarts de millièmes de seconde.

Notre cerveau nous permet alors d'estimer la profondeur des objets vus. Ce principe de vision 3D se base sur le fait qu'un objet plus proche n'a pas la même position entre les deux yeux, qu'un objet lointain qui lui varie moins entre les deux yeux. Ce principe appelé parallaxe fut utilisé tout d'abord en utilisant des stéréogrammes, qui sont des photos prises en parallèles et qui vues avec des lunettes spéciales permettent de voir la profondeur. Ce principe est toujours utilisé dans la capture de scène 3D comme il est possible de le voir dans de nombreux films sortis récemment.

Notre but est différent, il ne s'agit pas de recréer parfaitement la réalité mais d'extraire des mesures d'une scène 3D complexe. En réalité cette thèse a pour but

de créer un nouvel outil de mesure. Ces mesures devront pouvoir être faites sur des scènes représentant aussi bien une scène fixe, comme une plage de galet, ou une scène végétale, comme un arbre ou des buissons. Comme nous l'avons dit précédemment, reconstruire tous ces types de scènes vont nous imposer des contraintes sur le choix des algorithmes à utiliser.

Depuis, de nombreux autres capteurs et d'autres techniques ont permis la création de nuages points 3D. Il était donc nécessaire de mettre en opposition les techniques actuelles de production de nuages de points 3D et les contraintes liées à la problématique de la thèse :

méthode	coût	automatisation	matériel	déformable
interférométrie radar	×	✓	×	✓
time-of-flight	×	✓	×	×
LIDAR	×	✓	×	✓
shape from shading	✓	×	✓	×
structured light	×	✓	×	×
shape from focus	✓	×	✓	×
ouverture codée	✓	✓	×	✓
images multi-vues	✓	✓	✓	✓

TABLE 1 – tableau de l'état de l'art opposé aux contraintes liées à la problématique

On s'aperçoit dans le tableau que seules les approches basées sur un ensemble de photos d'une même scène apportent une réponse à toutes les contraintes. Les approches permettant la création de nuages de points 3D par multi-vues sont nombreuses. Nous pouvons séparer les approches en deux familles distinctes :

- La première famille cherche à se servir de l'information contenue dans les images pour retrouver la géométrie 3D [Hartley and Zisserman, 2004], [Faugeras, 1993]. Les approches de cette famille cherchent à mettre en correspondance les pixels des images. Cette information appelée disparité permet de connaître la profondeur. Une fois la géométrie de prise de vue connue, il est possible de produire un nuage de points 3D.

- La deuxième famille d'algorithmes utilise la géométrie de prises de vue pour produire un nuage de points. Dans ces approches, différents axes nous sont apparus.

Le premier axe utilise une discrétisation de l'espace 3D pour estimer une surface 3D en trouvant la surface minimisant une énergie [Kolmogorov and Zabih, 2002]. Cette énergie correspond souvent à l'intensité des reprojections des points dans les images.

Le problème de ces approches est qu'elles nécessitent une très bonne connaissance de la géométrie 3D. Or, pour obtenir la qualité nécessaire, il est préférable d'utiliser des cibles, ce qui est contraire à la contrainte d'automatisation. Aussi, cette approche n'a pas été choisie.

Le deuxième axe utilise des silhouettes pour "tailler" l'espace 3D [Hernández Esteban and Schmitt, 2004], [Franco and Boyer, 2005]. Ces techniques nécessitent en plus d'une très bonne connaissance de la géométrie,

une création de silhouettes allant à l'encontre des contraintes.

Enfin, le dernier axe utilise une information fragmentaire de la reconstruction 3D pour faire croître des plans à partir de ces points graines [Furukawa and Ponce, 2009]. Ces techniques, bien qu'offrant les meilleurs résultats dans le concours publié dans [Scharstein and Szeliski, 2002], n'offrent pas la densité de points nécessaires à l'extraction de paramètres de surfaces robustes.

Ainsi, seules les approches utilisant l'information des images pour reconstruire la 3D ont pu être utilisées. Nous expliquerons dans le premier chapitre la chaîne choisie en fonction du meilleur rapport qualité/contraintes, en cherchant les algorithmes les plus rapides.

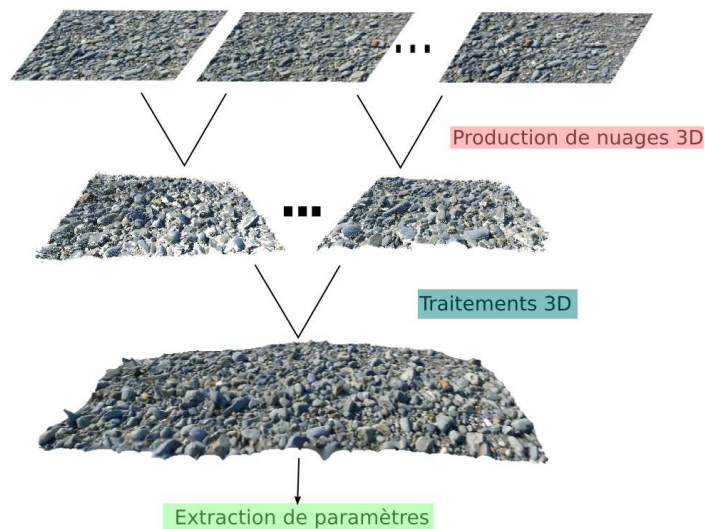


FIGURE 2 – Schéma des étapes pour aboutir au nuage final

Nous nous sommes aperçus que les résultats obtenus par ces méthodes n'étaient pas satisfaisantes pour des mesures, car elles comprenaient trop de point aberrants, de bruits et de redondances. Pour résoudre ces problèmes, nous avons dû créer une chaîne complète de suppression de bruit, chaîne expliquée dans le chapitre 2.

Enfin, une fois ces deux étapes effectuées, nous avons commencé à extraire des mesures.

Ces étapes sont résumées dans la figure 2.

Le manuscrit s'inscrit dans une démarche chronologique comme expliqué dans la figure 3, nous tenterons d'expliquer ainsi comment, à partir d'une série d'images prises sans contrainte, nous extrairons des informations.

D'abord, nous expliquerons tous les algorithmes utilisés pour produire, à partir d'un couple d'image, un nuage 3D. Nous commencerons par expliquer la mise en

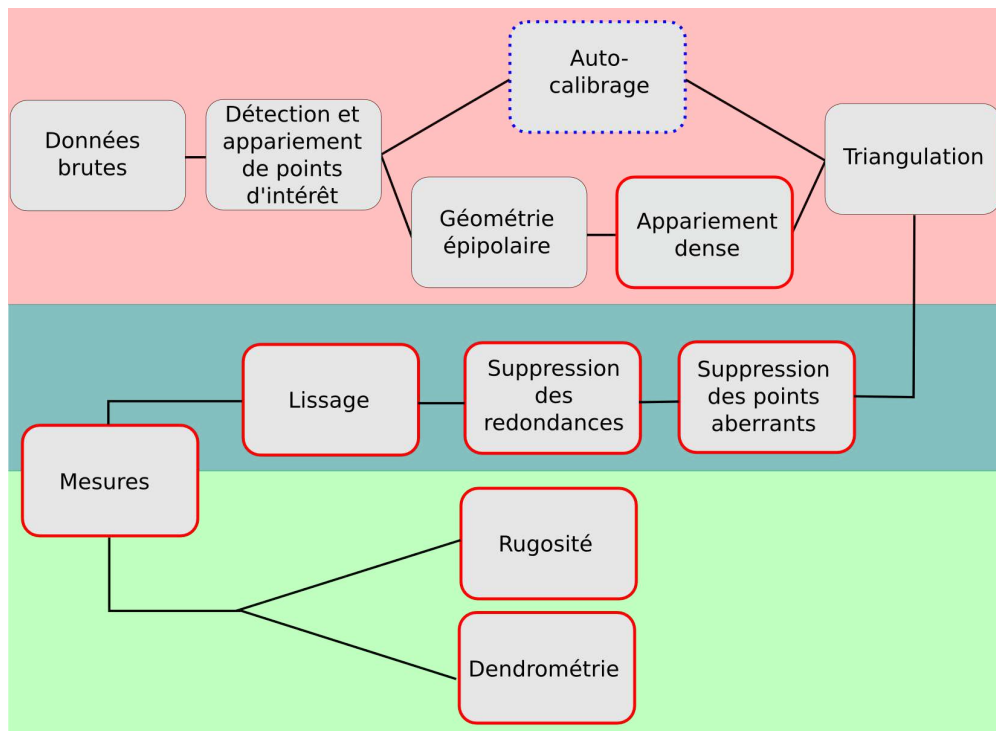


FIGURE 3 – Pipeline complet des algorithmes utilisés pour extraire des paramètres de surfaces de points 3D à partir de données brutes. Les carrés cerclés de rouges sont ceux sur lesquels nous avons pu apporter une amélioration.

correspondance de points saillants, c'est-à-dire que nous chercherons des points particuliers dans la première image que nous apparierons dans la deuxième image. A partir de ces correspondances, nous extrairons la géométrie épipolaire, géométrie décrivant l'espace du couple d'images qui nous servira ensuite à simplifier la mise en correspondance dense du couple d'images. Cette correspondance permet de connaître pour chaque pixel d'une image, à quel pixel de l'autre image il correspond. En parallèle de la géométrie épipolaire, un calibrage est effectué à partir des correspondances de points saillants. Ce calibrage s'effectue de manière automatique et nous permet de connaître les paramètres qui définissent le passage de l'espace image à l'espace 3D. Ensuite en se servant de ces paramètres de calibrage et des correspondances denses, une triangulation est effectuée permettant de récupérer des nuages 3D bruts.

Ensuite, nous expliquerons comment passer d'un certain nombre de nuages 3D, présentant du bruit et de la redondance, à un seul nuage de qualité. Nous présenterons donc un algorithme de filtrage des nuages de points par une étude de leurs voisinages. Cet algorithme utilise l'hypothèse qu'un point qui est mal placé est forcément loin des autres points bien placés. Ainsi, en étudiant le nombre de voisins, il est possible de savoir si un point est bien placé. Ensuite la redondance sera supprimée par un algorithme utilisant à la fois des contraintes géométriques et colorimétriques. Le nuage produit sera alors lissé pour supprimer le bruit résiduel en utilisant des techniques

récentes de lissage par moindres carrés glissants.

Une fois le nuage 3D trouvé nous étudierons sa véracité aux moyens de comparaisons entre mesures réelles et mesures 3D. Cette validation a pour but de prouver l'efficacité et la robustesse de l'ensemble de la méthode. Cette validation aboutira sur une étude de la provenance des erreurs. A partir de ces études, nous préconiserons un ensemble de conseils permettant aux non-spécialistes de connaître la méthode permettant d'obtenir la meilleure reconstruction 3D possible.

Nous concluons ce manuscrit par les paramètres de surfaces, leur utilisation, leur extraction et leur validation. Dans ces paramètres nous retrouverons les paramètres de rugosité, complexes à définir, et pour lesquelles nous apportons une vraie nouveauté, c'est-à-dire le calcul sur des surfaces et non sur des profils. Enfin nous aborderons l'extraction des paramètres dendrométriques, correspondants aux paramètres définissant les végétaux. Nous montrerons surtout comment nous avons pu extraire les informations des nuages de points 3D. Nous finirons par un état des lieux des perspectives à venir notamment des mesures complexes qui auraient pu être extraites.

Chapitre 1

Reconstruction 3D

L'idée générale de la thèse visait à extraire des mesures d'un nuage de points 3D. La première étape était donc de créer ces nuages de points 3D à partir de photos. Les contraintes, que nous imposent l'utilisation de ces outils par des non-spécialistes, nous astreignent à n'utiliser que des images prises avec un minimum d'obligations. Or, reconstruire l'information 3D à partir de photos est un sujet très étudié depuis une trentaine d'année. Nous avons fait un choix entre les deux approches principales. La première reconstruit l'information à partir d'une discrétisation de l'espace 3D en reprojectant les points 3D dans les images. Cette approche nécessitant un calibrage fiable, nous avons choisi, comme nous l'avons expliqué dans l'introduction, l'approche projetant des points mis en correspondance. Cette approche satisfait à toutes les contraintes citées dans l'introduction.

Dans cette approche, les algorithmes les plus classiques sont donnés par les livres [Hartley and Zisserman, 2000], [Faugeras, 1993] et [Pollefeys et al.,], et ont pour point commun de tous utiliser la géométrie épipolaire, qui définit l'espace d'un couple d'images. Cette géométrie est parfaitement expliquée par la matrice fondamentale calculée, soit par extrapolation des paramètres de calibrage, soit de manière analytique en se servant de points de correspondances entre deux images. Or, comme nous pouvons le voir dans la figure 1, nous effectuerons un auto-calibrage pour satisfaire aux contraintes d'automatisation maximum. Ce type de calibrage n'apporte pas la précision nécessaire pour extraire la géométrie épipolaire correctement. Ainsi, nous avons fait le choix d'estimer les paramètres de la géométrie épipolaire à partir de correspondances.

Nous expliquerons dans ce chapitre les algorithmes choisis. Ces algorithmes forment une chaîne d'algorithmes, dont nous tenterons d'expliquer les choix. Ces choix sont généralement conduits par la volonté de faire la chaîne d'algorithmes la plus robuste aux contraintes expliquées dans l'introduction.

Dans cette section nous développerons les différents algorithmes utilisés pour extraire l'information 3D des images. Nous allons expliquer chaque étape de la figure 1. Le but est d'obtenir, à partir de photos prises avec un minimum de contraintes, un nuage de points 3D sur lequel il est possible d'extraire des informations biologiques ou géo-physiques.

Nous traiterons ici des étapes qui permettent de créer un nuage de points 3D pour chaque couple d'images traité, et de positionner l'ensemble de ces nuages de points dans une même géométrie.

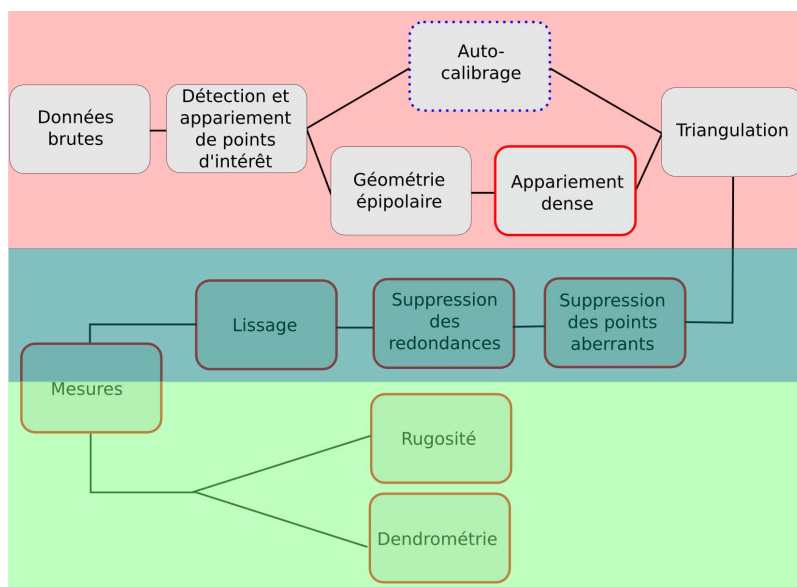


FIGURE 1 – Chaîne de traitement général pour la reconstruction d'un nuage 3D

Nous développerons tout d'abord les aspects concernant la qualité des données souhaitées, ensuite nous parlerons de la détection et de la mise en correspondance de points saillants permettant la mise en place d'une procédure d'auto-calibrage. Enfin, l'étude de la géométrie épipolaire, définie par la matrice fondamentale, et le calcul de carte de disparité nous permettront de créer un nuage de points 3D par triangulation.

1.1 Données

Les images brutes utilisées ont été acquises avec un appareil photo¹ du commerce et enregistrées au format "jpg". Sur celles-ci, aucun pré-traitement n'a été effectué.

Les seules contraintes imposées sont donc de prendre des photos de la même scène, a fortiori avec à peu près les mêmes conditions d'éclairage, non floues et surtout pour lesquelles il est possible de connaître les paramètres internes de prise de vue.

1.1.1 Modèle projectif

L'appareil photo, ou la caméra, se caractérise par un modèle d'acquisition à "tête d'épingle" (pinhole), ou modèle sténopé, qui est décrit par un ensemble de

1. Panasonic Lumix DMC-FZ50 ou Pentax K10D

paramètres de calibrage, qui permettent alors de connaître parfaitement le modèle projectif comme présenté dans la figure 1.1.1, composé des paramètres intrinsèques et extrinsèques [Mcglone et al., 1980].

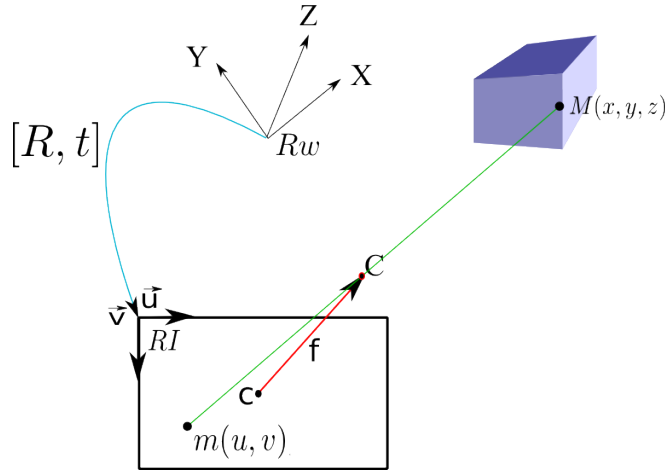


FIGURE 1.1.1 – Schéma du modèle de prise de vue projective avec $RI[\vec{u}\vec{v}]$ le repère image, et $Rw[X, Y, Z]$ le repère euclidien ou repère monde

Le modèle projectif se décrit de manière analytique tel que :

$$m_{i,j} = P_j \times M_i \quad (1.1)$$

où M_i est le point dans le repère réel (ou monde), $m_{i,j}$ le projeté du point M_i dans le repère de l'image, c'est-à-dire un pixel de l'image et P_j la matrice de projection de la caméra décrivant les paramètres intrinsèques et extrinsèques tel que :

$$P = K \times [R \ t] \quad (1.2)$$

où $[R \ t]$ les paramètres extrinsèques, avec R la matrice 3x3 de rotation et t le vecteur 3x1 de translation et K correspond à la matrice 3x3 des paramètres intrinsèques :

$$K = \begin{bmatrix} f_c & \gamma & c_c \\ 0 & f_l & c_l \\ 0 & 0 & 1 \end{bmatrix} \quad (1.3)$$

Avec $f_c = f_l = f$ la longueur focale car on supposera les pixels carrés, γ le paramètre traduisant le caractère non-orthogonal des cellules CCD de la caméra, ce paramètre est généralement proche du nul et sera donc négligé, $[c_c, c_l]$ le point principal correspondant à la projection orthogonal du centre optique sur l'image et considéré comme le centre de l'image. Le centre optique C est le point de l'espace par lequel passe tous les rayons lumineux. Dans notre chaîne de traitement, nous considérerons que les images sont enregistrées avec des méta-données : la longueur focale, la taille du capteur CCD, ainsi que si possible les paramètres de distorsions. Ces informations sont généralement associées à chaque photographie numérique et sont appelées EXIF.

1.1.2 Modèle stéréoscopique

Le modèle projectif nous permet d'introduire le modèle stéréoscopique, c'est-à-dire la géométrie définissant une paire d'image. Le principe de disparité repose sur le fait qu'il existe une relation entre l'espacement pixélaire entre deux images et sa profondeur comme on peut le voir sur la figure 1.1.2.

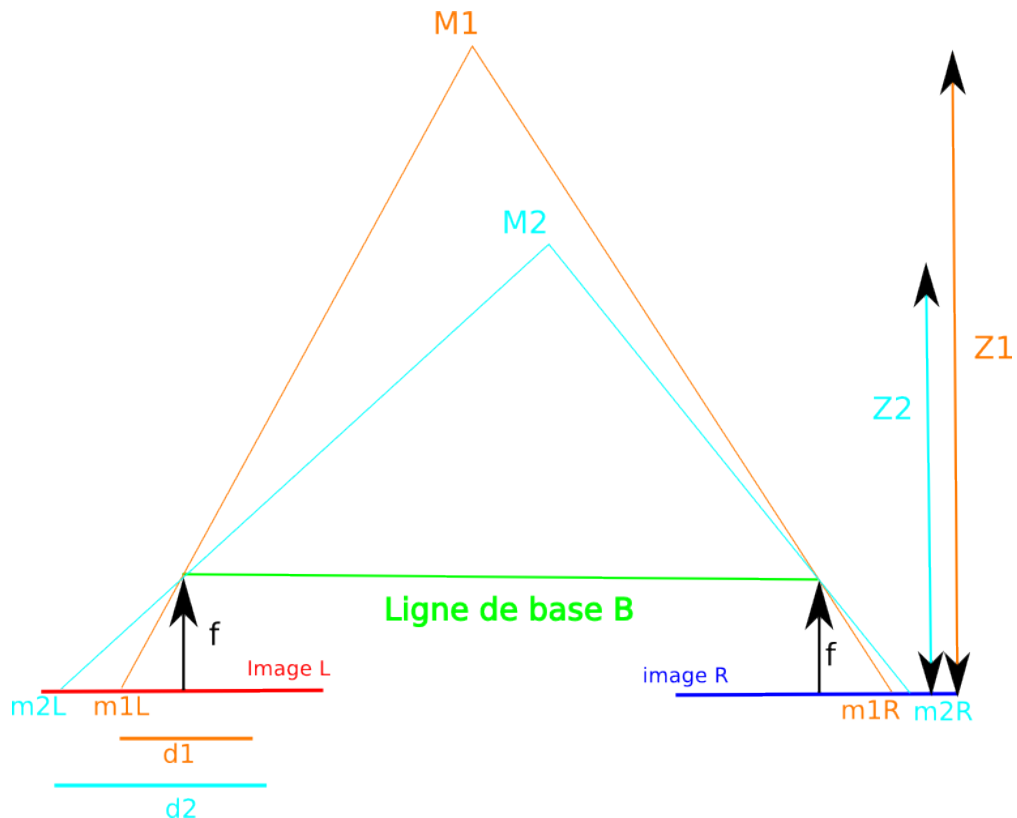


FIGURE 1.1.2 – Schéma de la vue stéréoscopique, avec deux points M1 et M2 se projetant dans les deux images

Sur la figure 1.1.2, la première disparité "d1" est plus petite que la deuxième "d2". C'est sur ce phénomène que repose toute la stéréovision, en effet si d_1 est plus petite que d_2 , c'est que le point M1 est plus éloigné que le point M2. La disparité est donc inversement proportionnelle à la distance du point à l'image. Or par la théorie des triangles semblables, on sait que pour un point M :

$$\frac{B}{Z} = \frac{B-d}{Z-f} \Leftrightarrow Z = \frac{B \times f}{d}$$

Ici la disparité d est considérée sur nos images rectifiées comme la différence des abscisses des points appariés, $d = x_r - x_l$.

1.1.3 Précision des mesures de profondeurs souhaitées

Il est important de connaître la précision en Z souhaité pour connaître le meilleur rapport B/H possible et ainsi imposer une manière de faire les prises de vues. Il est possible selon les travaux de [Saleh and Scarpace, 2000] de fixer certains paramètres de prises de vue en fonction de la précision en Z souhaitée :

$$\Delta Z = \rho \varepsilon \frac{H^2}{fB} \quad (1.4)$$

avec ρ la somme algébrique proportionnelle des erreurs, et selon [Saleh and Scarpace, 2000] généralement correspondant à 0.3 ou 0.6, ε l'erreur du capteur, c'est-à-dire de la lentille, valeur contenue dans les EXIF (dans notre cas $20 \mu\text{m}$), H la hauteur de prise de vue, f la longueur focale, B la base de prise de vue, c'est-à-dire la distance qui sépare les deux prises de vues stéréoscopiques. En considérant le cas défavorable d'une valeur de ρ à 0.6 et en considérant qu'avec des prises de vue faites par une personne à la verticale la hauteur ne peut dépasser deux mètres, ainsi qu'une longueur focale que l'on considérera fixe à 28 mm entre deux prises de vue, seul B peut varier pour obtenir la résolution en z souhaitée.

Influence du rapport B/H

Si une précision au millimètre en z est souhaitée, il est montré dans la figure 1.1.3 l'évolution de la longueur de base pour différentes hauteurs de prises de vue fixées :

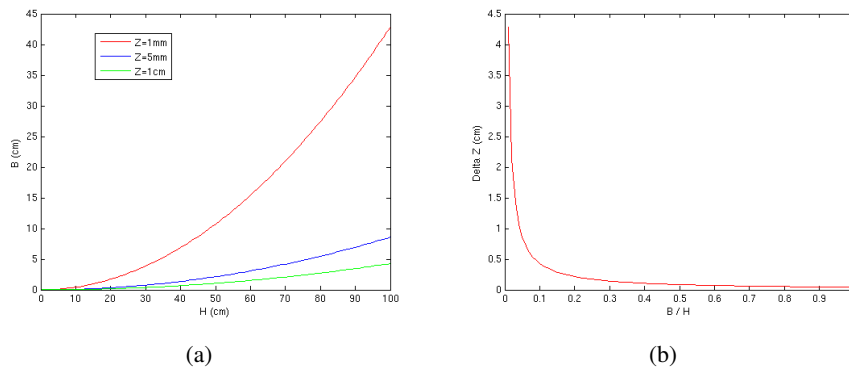


FIGURE 1.1.3 – (a) évolution de la longueur de base en fonction de l'évolution de la hauteur de prise de vue entre 0 et 100 centimètres avec une résolution en Z fixe, (b) évolution de la résolution en Z en fonction du rapport B/H

Un rapport B/H élevé permet une reconstruction efficace, mais diminue la surface de recouvrement. A l'inverse, un rapport B/H bas augmente la surface de recouvrement mais diminue la résolution en Z .

Les expériences menées ont montré qu'un certain nombre de contraintes supplémentaires sont nécessaires à l'obtention de mesures plus précises :

- les images ne doivent pas être floues ;

- avoir le minimum de surfaces uniformes (un mur blanc par exemple) dans la scène à reconstruire ;
- avoir le minimum de structures redondantes comme un radiateur ;
- respecter la règle de B / H expliquée ci-avant ;

1.2 Appariement de points d'intérêt

Le premier maillon de la chaîne de traitement de stéréovision est l'étape de mise en correspondance de points particuliers, dit saillants. Cette étape va permettre ensuite de connaître la géométrie épipolaire du couple d'image et d'auto-calibrer le système. Pour l'appariement de points, trois étapes sont nécessaires :

- La première étape est de détecter des structures géométriques particulières apparaissant dans les images. Il existe différentes structures détectables dans les images, les points saillants, les régions, la texture ou les contours, dont on estime qu'elles se répéteront dans l'autre image. Les structures les plus simples et souvent les plus robustes à mettre en correspondance sont les points saillants.
- La deuxième étape est de décrire les points par rapport à leurs voisinages. C'est-à-dire d'extraire des informations permettant de rendre robuste le critère d'appariement à différentes transformations géométriques, soit par le calcul de coefficients invariants, soit par la normalisation du voisinage selon des critères géométriques.
- Enfin la dernière étape est la mise en correspondance de ces descripteurs pour apparier les points particuliers.

1.2.1 Détections

De nombreux détecteurs de points d'intérêt ont été développés. Si certains utilisent une différence d'images de dérivées en x et y comme le sont les détecteurs de coins de type "Harris" [Harris and Stephens, 1988] qui ont été longtemps utilisés, d'autres détectent les points saillants par un calcul des maxima du Laplacien normalisé à l'échelle sur une suite d'images à différentes échelles, repérant ainsi l'échelle caractéristique. L'intérêt est de rendre les points invariants par échelle en définissant qu'un point saillant est un extremum local de la représentation en espace-échelle.

Dans cette deuxième catégorie, le plus connu des détecteurs est l'algorithme SIFT (scale invariant features transform) [Lowe, 1999]. Cet algorithme cherche une approximation du Laplacien gaussien normalisé par une différence de gaussienne (DoG), car le calcul du Laplacien est lent et coûteux en ressources mémoire.

Une autre approximation est de réduire l'espace-échelle à une pyramide d'images sous-échantillonnées d'un facteur 2. D'autres détecteurs utilisent les idées directrices de l'algorithme SIFT et en améliorent la qualité et la rapidité. Par exemple l'algorithme SURF (Speed-Up Robust Features) [Bay et al., 2008] améliore la rapidité de l'algorithme en se basant sur un calcul du déterminant du Hessien. Il est possible d'améliorer la robustesse de l'algorithme en permettant de rendre les points invariants aux transformations affines par une estimation des transformations affines. Ces points appelés ASIFT sont décrits dans [Morel and Yu, 2009].

1.2.2 Descripteurs

La deuxième étape consiste à décrire le voisinage de chaque point détecté au travers d'un certain nombre de paramètres. Une description simple est la valeur des pixels voisins. Les algorithmes SIFT, ASIFT utilisent quant à eux le calcul d'histogrammes d'orientations locales du gradient à l'échelle caractéristique autour de chaque point trouvé, rendant ainsi les points trouvés invariants par rotation dans un plan 2D. L'algorithme SURF calcule leurs descripteurs comme la somme des réponses en ondelettes de Harr du voisinage, rendant les descripteurs plus robustes aux bruits et aux rotations, et surtout plus rapides car il est très simple de faire la projection en ondelettes grâce à l'image intégrale utilisée depuis la formulation de [Viola and Jones, 2002].

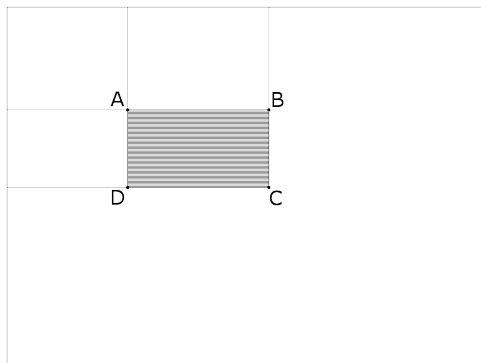


FIGURE 1.2.1 – Shéma de l'image intégrale.

Une image intégrale étant une image où chaque pixel correspond à la somme des pixels en haut à gauche du pixel, comme montré dans la figure 1.2.1. Pour un pixel $p(x,y)$ sa valeur dans l'image intégrale est :

$$I(x,y) = \sum_{i=0}^x \sum_{j=0}^y p(i,j)$$

Une fois l'image intégrale calculée, trouver la somme des pixels de n'importe quel rectangle ABCD, notamment pour la projection en ondelette de Harr, se calcule facilement par une simple addition :

$$\sum_{i \in [x_A, x_C]} \sum_{j \in [y_A, y_C]} p(i,j) = I(A) + I(C) - I(B) - I(D) \quad (1.5)$$

L'algorithme SURF découpe le voisinage en seize zones sur lesquelles les sommes $\sum I_x$ des dérivées horizontales, $\sum |I_x|$, $\sum I_y$ et $\sum |I_y|$ sont calculées et placées dans 64 descripteurs au lieu de 128 pour SIFT et ASIFT.

1.2.3 Appariements

La mise en correspondance de ces points est basée sur une distance euclidienne ou une distance de Mahalanobis entre les valeurs des descripteurs locaux.

C'est-à-dire qu'en considérant les vecteurs de descripteurs en chaque point saillant : $V_l = \{l_1, l_2, \dots, l_n\}$ et $V_r = \{r_1, r_2, \dots, r_n\}$, la distance entre ces deux vecteurs de dimension n est calculée telle que $d(V_l, V_r) = \sqrt{(l_1 - r_1)^2 + (l_2 - r_2)^2 + \dots + (l_n - r_n)^2}$. D'autres techniques existent comme un appariement par des méthodes dites "a contrario" [Rabin et al., 2009] et ont montré de très bons résultats.

Notre problématique étant la mise en correspondance d'images prises sans contrainte, l'algorithme ASIFT semble le plus indiqué mais les ressources en mémoire et le temps de calcul sont des freins à son utilisation sur le terrain pour l'obtention rapide de données. Ainsi SURF a été préféré dans cette approche.

De plus, dans les résultats ci-dessous, les images ne peuvent être exploitées directement lorsqu'elles dépassent une certaine taille par l'algorithme SIFT car elles prennent trop de place mémoire. Ainsi, pour pouvoir contenir la pyramide d'image, les images lorsqu'elles dépassent un million de pixels sont mosaïquées pour SIFT et sous-échantillonnées pour ASIFT. Au contraire, l'algorithme SURF supporte bien les grosses images et peut être utilisé directement, ceci est lié au fait que l'algorithme utilise des images intégrales pour approcher grossièrement la pyramide d'échelles.

1.2.4 Résultats et discussions



FIGURE 1.2.2 – Points SIFT sur les images de la mosquée de Damas.

Malgré les précautions prises pour éviter les mauvais appariements, que nous appellerons "outliers", certains subsistent. La raison qui peut expliquer la présence de ces mauvais appariements est la présence de transformations projectives entre nos images, qui ne sont pas prises en compte dans les descripteurs. La suppression de ces outliers est effectuée par un algorithme de RANSAC (RANdom SAMple Consensus) [Fischler and Bolles, 1981] ayant montré sa robustesse depuis de nombreuses années. Pour tester la fiabilité des algorithmes d'appariement de points saillants, on retrouvera le tableau récapitulatif des différentes caractéristiques 1.1 et 1.2, montrant la robustesse des différents algorithmes expliqués dans ce chapitre.

Les tableaux nous montrent que l'algorithme SURF obtient moins d'appariement que l'algorithme ASIFT. Mais, malgré un nombre d'appariement souvent moins bon



FIGURE 1.2.3 – Points ASIFT sur les images de la mosquée de Damas.



FIGURE 1.2.4 – Points SURF sur les images de la mosquée de Damas.

Méthode	nombre de points appariés	nombre de bons appariements	temps (en sec)
SIFT	38	25	3
ASIFT	128	32	53
SURF	51	27	<1

TABLE 1.1 – Résultats des appariements sur l'image de la mosquée (836*556).

Méthode	nombre de points appariés	nombre de bons appariements	temps (en sec)
SIFT	5013	4746	35
ASIFT	11233	9030	159
SURF	3385	2878	8

TABLE 1.2 – Résultats des appariements sur l'image de galets (3000*2000).

que l'algorithme ASIFT, l'algorithme SURF obtient des résultats utilisables dans le reste de notre chaîne de traitement et de manière beaucoup plus rapide. Ces tests nous ont conforté dans l'idée du choix de SURF comme apparieur de points saillants, offrant le meilleur compromis rapidité fiabilité.

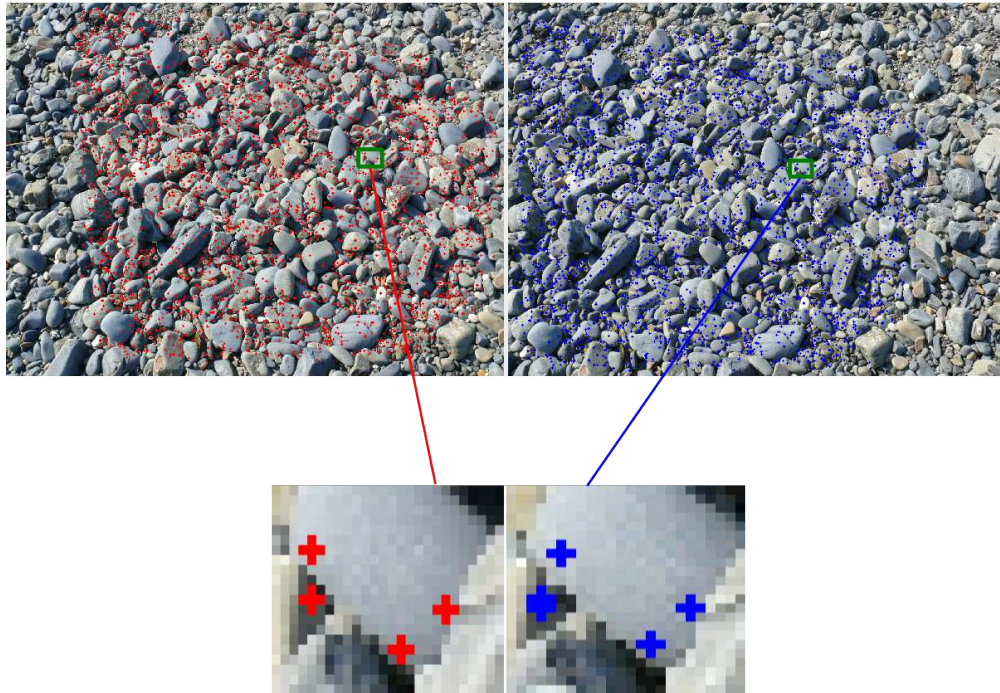


FIGURE 1.2.5 – Points SURF sur les images de galets et détails de points correspondants.

Méthode	nombre de points trouvés	nombre de bons appariements	temps (en sec)
SIFT	-	-	-
ASIFT	125	25	45
SURF	25	11	1

TABLE 1.3 – Résultats de l'appariement sur le couple d'images de cailloux (1824x1216), de la figure 1.2.6 ((a) et (b)). On peut remarquer que SIFT ne détecte aucune correspondance à cause de la transformation entre les deux images.

Méthode	nombre de points trouvés	nombre de bons appariements	temps (en sec)
SIFT	19	5	15
ASIFT	95	20	44
SURF	20	10	2

TABLE 1.4 – Résultats de l'appariement sur le couple d'images de cailloux (1824x1216), de la figure 1.2.6 ((a) et (c)), sur lesquels on observe que le nombre de points trouvés reste insuffisant pour la suite de la chaîne de traitement.

1.3 Géométrie épipolaire

Une fois la mise en correspondance de points particuliers effectuée, une étape d'appariement dense est effectuée pour tirer un maximum d'information des couples d'images. Le problème est que la recherche s'effectue dans un espace 2D, c'est à dire celui

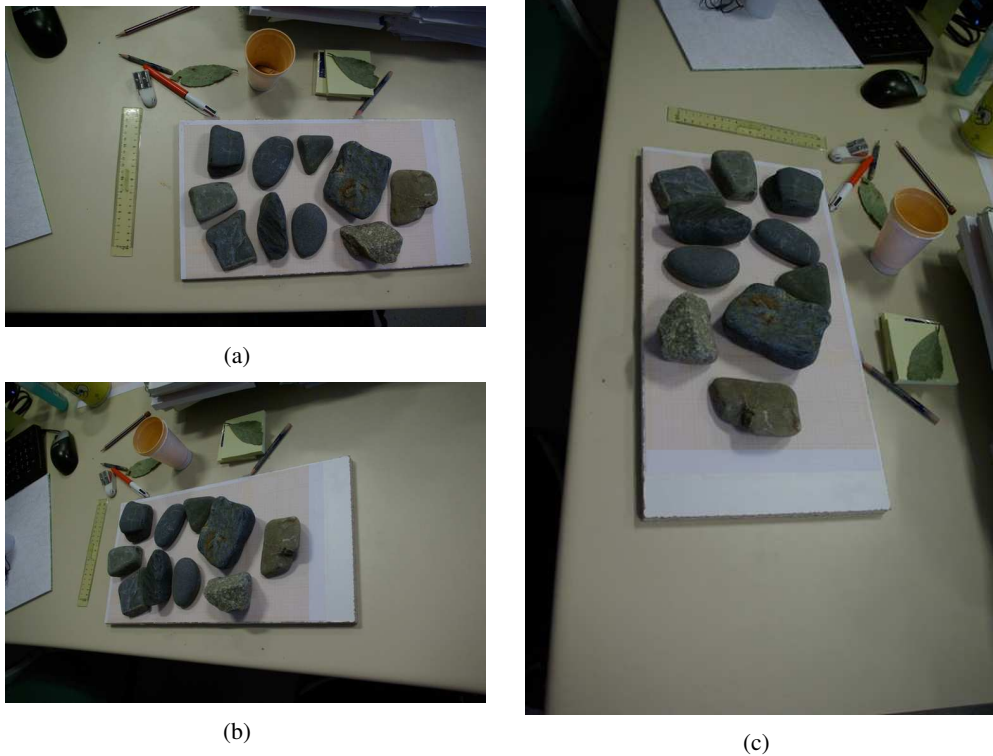


FIGURE 1.2.6 – (a) Image de cailloux témoin avec laquelle sera fait l'appariement test, (b)(c) images servant aux tests de robustesse.

de l'image, or cette recherche est trop longue et pas assez robuste pour pouvoir être effectuée.

Pour simplifier cette recherche, il est montré qu'il est possible d'effectuer la recherche d'un correspondant, d'un pixel de l'image gauche, seulement sur un segment de l'image droite, et non sur l'image dans son ensemble. Cette possibilité est expliquée par l'équation fondamentale de la géométrie épipolaire [Longuet-Higgins, 1987], définissant la matrice fondamentale :

$$m_r^T F m_l = 0 \quad (1.6)$$

avec m_l un point de l'image gauche, m_r son correspondant, m_r^T le transposé de m_r et F la matrice fondamentale de rang 2.

La matrice fondamentale explique ainsi parfaitement la géométrie épipolaire entre les deux images. Les techniques pour la calculer utilisent la connaissance de correspondances entre images gauche et droite. Les points d'intérêt appariés nous permettent de trouver la matrice fondamentale [Hartley and Zisserman, 2004], [Faugeras, 1993] en résolvant un système d'équations grâce au "8-points algorithm" ou "7-points algorithm"[Hartley and Zisserman, 2004].

La géométrie épipolaire associe à chaque couple de points m_l et m_r leurs lignes épipo-

lignes l_l et l_r définies telles que :

$$l_r \simeq F m_l$$

$$l_l \simeq m_r^T F$$

Elle définit aussi la position des épiques e_l et e_r comme points d'intersections de toutes les lignes épipolaires d'une image :

$$F e_l = 0$$

$$e_r^T F = 0$$

Enfin, la contrainte de coplanarité épipolaire définit que les points m_l, m_r ainsi que leurs lignes épipolaires associées et les deux épiques sont coplanaires. Cette contrainte permet la recherche d'un correspondant de l'image gauche dans l'image droite le long de la ligne épipolaire associée et non plus sur toute l'image.

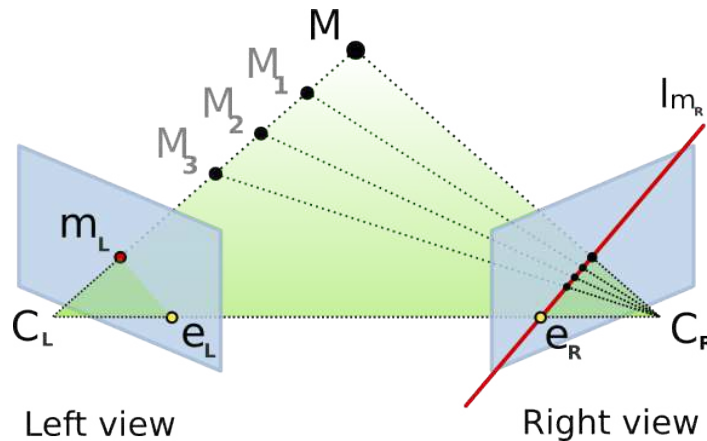


FIGURE 1.3.1 – Schéma de la géométrie épipolaire d'une paire d'images stéréo, e_l et e_r respectivement les épiques gauche et droite, O_l et O_r les centres optiques gauche et droit, et à X_l correspond le point X_r trouvé sur la ligne épipolaire en rouge sur l'image droite.

Pour une simplification de recherche de correspondants, il fut choisi de rectifier les lignes épipolaires, les rendant parallèles les unes aux autres. Cette étape permettra une recherche directement le long de la ligne de pixels.

1.3.1 Rectification épipolaire

La rectification épipolaire permet de placer dans le même plan les deux images en contraignant la rectification pour que les lignes épipolaires soient parallèles et pour que chaque pixel trouve son correspondant sur la ligne de pixel de l'autre image, principe résumé sur la figure 1.3.2

Les images sont rectifiées le long des lignes épipolaires pour que la recherche s'effectue le long d'une ligne de pixels. Ce qui entraîne un positionnement des épiques à l'infini. Différents algorithmes existent :

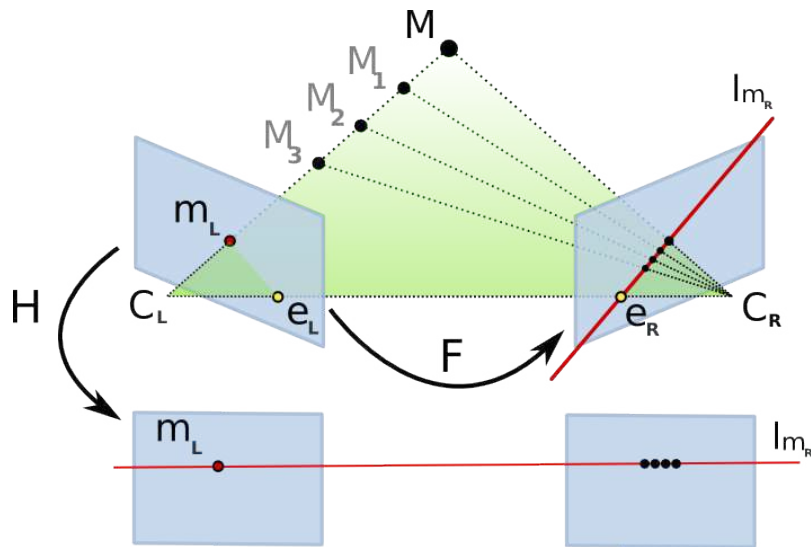


FIGURE 1.3.2 – Schéma de la rectification épipolaire, ainsi une fois rectifiées, les deux images seront plus aisément mises en correspondances.

- les algorithmes linéaires [Hartley and Zisserman, 2004],[Faugeras, 1993] dans lesquels un plan parallèle à la ligne de base, reliant les deux centres optiques, est cherché et qui minimise les distorsions.
- les algorithmes polaires [Oram, 2001], [Pollefeys et al., 1999] dans lesquels on cherche à rendre les lignes parallèles en paramétrisant les images en coordonnées polaires centrées autour des épipoles.
- les algorithmes se servant des données de calibrage [Fusiello et al., 2000]. La robustesse de ce type d’approche étant équivalente à celle du calibrage.

Pour des raisons expliquées dans le chapitre suivant, les résultats obtenus par auto-calibrage contiennent des erreurs, nous faisant choisir parmi les algorithmes n’utilisant pas les données de calibrage. Utiliser ou non les données de calibrage permet une rectification épipolaire équivalente. Ainsi, ne pas se servir du calibrage nous semblait légitime.

Pour la rectification sans données de calibrage, nous avons préféré une rectification polaire [Oram, 2001] plutôt qu’une rectification linéaire [Hartley and Zisserman, 2004], qui entraîne généralement une plus grande distorsion des images. La rectification linéaire possède tout de même de nombreuses qualités, une simplicité algorithmique, une rapidité d’exécution ainsi qu’une préservation des éléments géométriques de l’image, les droites par exemple. Mais la rectification linéaire est principalement adaptée à des couples d’images prises à partir de deux caméras proches l’une de l’autre, et non comme dans notre cas à des images prises à la main sans contraintes, entraînant sur les images rectifiées de trop nombreuses distorsions. Notre choix s’est donc porté sur une rectification polaire. La rectification épipolaire de D. Oram [Oram, 2001] a été préférée car en utilisant la même transformation non-linéaire sur les deux images, le problème des

points appariés déformés différemment entre les deux images est évité. De plus l'implémentation montre une simplification par rapport à l'approche proposée par [Pollefeys et al., 1999].

La rectification sera effectuée en deux temps, en calculant tout d'abord la matrice de transformation, appelée homographie, permettant de passer des lignes épipolaires d'une image aux lignes épipolaires de l'autre, puis une reprojection des pixels en coordonnées polaires.

Choix de l'homographie compatible

Dans un premier temps, une matrice d'homographie compatible entre les deux images est estimée, celle-ci doit faire coïncider les lignes épipolaires et non les rendre parallèles comme dans une rectification linéaire. Cette matrice H doit permettre une cartographie entre les points correspondant m_l et m_r , telle que $m_l \simeq Hm_r$. Il est montré dans [Oram, 2001] que cette matrice H , est calculée comme :

$$H \simeq [e_r]_{\times} F - e_r a^T \quad (1.7)$$

où e_r est l'épipoles de la deuxième image, a^T le transposé d'un vecteur arbitraire de tel sorte que $\det(H) \neq 0$ et F la matrice fondamentale. Ainsi, la meilleure transformation, plaçant les deux images dans le meilleur plan, sera celle qui rapprochera le plus les projetés des points m_l et m_r par H . Ainsi pour trouver cette transformation il suffit de minimiser le critère en utilisant les résultats de l'homographie compatible en 1.7 :

$$\min_H \sum_{i=1}^n d(m_{l,i}, H m_{r,i}) \quad (1.8)$$

en remplaçant H par l'équation 1.7 :

$$\min_H \sum_{i=1}^n d(m_{l,i}, ([e_r]_{\times} F - e_r a^T) m_{r,i})$$

où d est la distance euclidienne.

En substituant la distance euclidienne par une distance algébrique et $[e_r]_{\times} F$ dans H , on obtient deux équations linéaires en terme "a" pour chaque couple de points appariés :

$$(m_{r_1} e_{r_3} - e_{r_1}) m_l^T a = (m_{r_1} h_3^T - h_1) m_l \quad (1.9)$$

$$(m_{r_2} e_{r_3} - e_{r_2}) m_l^T a = (m_{r_2} h_3^T - h_2) m_l \quad (1.10)$$

où les indices correspondent à la i ème valeur du vecteur, sauf pour les indices de h qui correspondent au numéro de la colonne. Ce système peut ainsi être résolu par moindres carrés. L'homographie compatible est ainsi estimée.

Mais, même si les valeurs sont conformes à la géométrie épipolaire, elles peuvent être incorrectes, voilà pourquoi une méthode d'échantillonnage aléatoire est utilisée pour supprimer ces valeurs.

Une fois H trouvé, elle est appliquée à tous les points de la deuxième image, pour les placer dans le plan de la première image. Une fois cette étape effectuée, il faut rectifier les images placées dans le même plan en utilisant les coordonnées polaires.

Projection polaire

Ainsi après avoir appliquée à l'image droite l'homographie compatible, une rectification visant à paralléliser les lignes épipolaires peut être effectuée. Pour cela, chaque point à rectifier est paramétrisé en terme de coordonnées polaires centrées à l'épipole. Pour paramétriser les coordonnées, il est nécessaire en chaque point de trouver :

- un angle polaire $\theta \in [-\pi, \pi]$;
- un rayon r correspondant à la distance positive du point rectifié à l'épipole ;

Mais avant qu'une paramétrisation ne soit possible, il faut d'abord calculer les régions communes aux deux images. En supposant l'image rectangulaire, ces régions trouvent leurs limites pour les coins de l'image. La plage d'angles polaires engendré par l'image doit être au plus de π . Donc, les coins maximaux seront trouvés en déterminant l'angle polaire en chaque coin et en sélectionnant les coins maximaux et minimaux. Ce choix sera effectué en calculant la plage totale des angles polaires et en choisissant celle inférieure à π . Ensuite, la même procédure est appliquée à la deuxième image. Enfin, la région commune est déterminée en appliquant ces angles extrêmes dans les images opposées comme expliquée dans la figure 1.3.3.

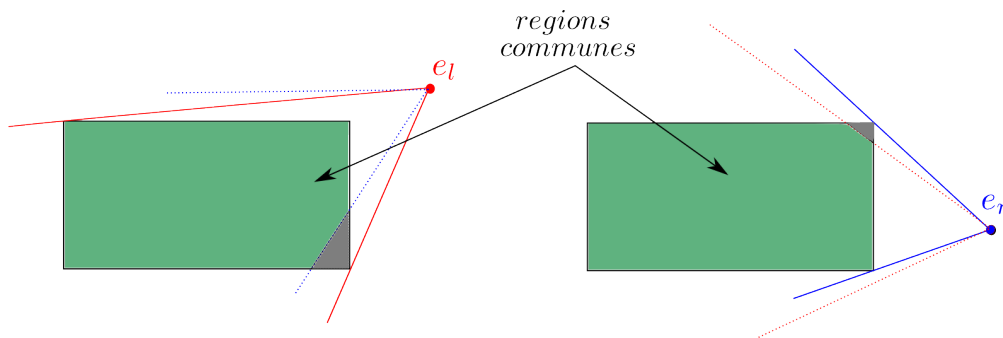


FIGURE 1.3.3 – Choix des régions communes grâce aux lignes épipolaires extrêmes.

Une fois la région commune trouvée, l'image sera rectifiée ligne épipolaire par ligne épipolaire. La distance minimum entre chaque ligne épipolaire sélectionnée doit être choisie avec soin pour éviter la compression des pixels. C'est à dire que deux lignes épipolaires qui seront consécutives sur l'image rectifiée doivent être choisies avec un certain θ tel qu'aucun pixel de l'image originale :

- ne soit pas pris en compte si l'angle est trop grand ;
- ne soit pas compressé si l'angle est trop petit ;

Ainsi les lignes épipolaire seront sélectionnées avec un angle évitant la perte et la compression des pixels, entre les lignes épipolaires consécutives. Pour cela, on choisira l'angle θ tel que :

$$\theta = \arctan\left(\frac{1}{d}\right) \quad (1.11)$$

avec d la distance minimale entre l'épipole et le point d'intersection entre la ligne épipolaire et le bord de l'image (voir la figure 1.3.4).

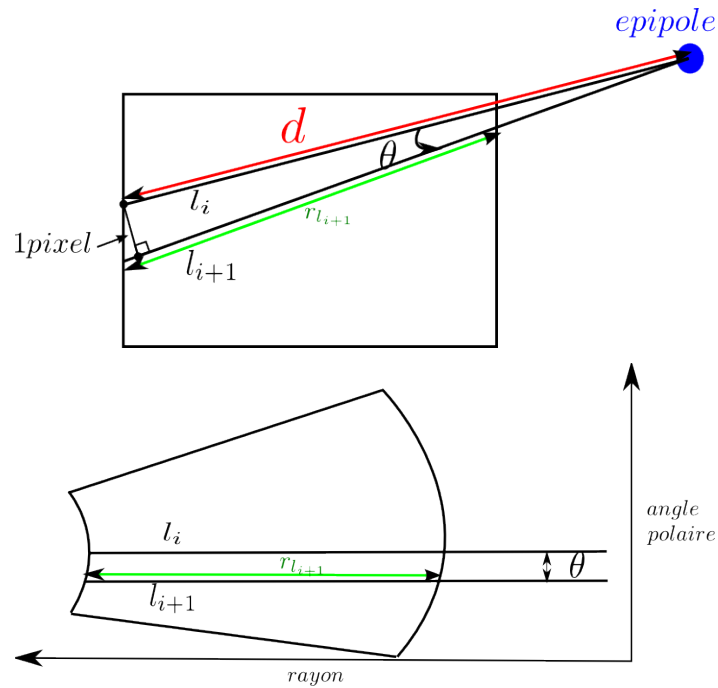


FIGURE 1.3.4 – Rectification épipolaire, avec le choix de l’angle θ et la paramétrisation en coordonnées polaires.



FIGURE 1.3.5 – Rectification épipolaire sur un couple d’images de galets. On s’aperçoit ici que l’angle choisi peut inverser le sens des images rectifiées, car il dépend de la position de l’épipole.

1.4 Appariement dense

Une fois les deux images rectifiées en épipolaire, une corrélation dense peut être effectuée pour qu’à chaque pixel de l’image gauche soit déterminé son correspondant dans l’image droite. Ce correspondant sera cherché grâce à la rectification épipolaire sur la même ligne de pixels.

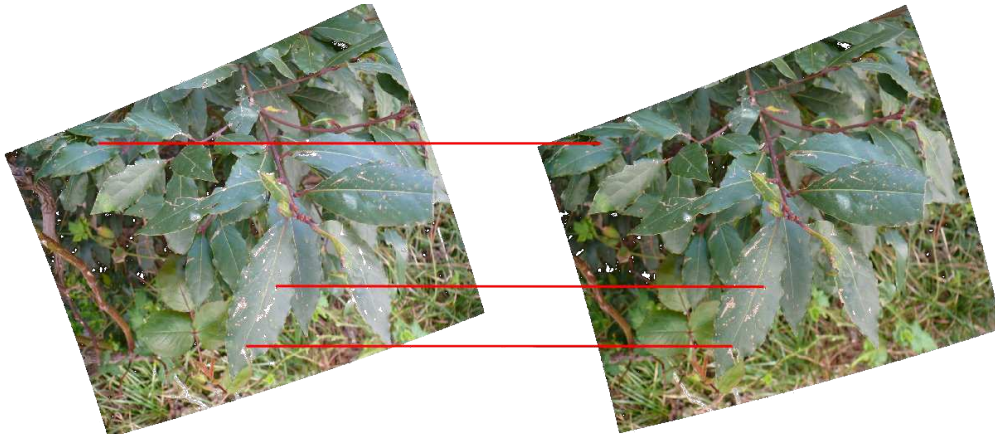


FIGURE 1.3.6 – Rectification épipolaire sur un couple d'images de feuilles avec en rouge des exemples de correspondances.

Dans cette partie, nous tenterons d'expliquer comment nous retrouvons la carte de disparité qui est l'image des disparités en chaque pixel. Le problème est donc de faire correspondre à chaque point de l'image gauche, son correspondant dans l'image droite. Une fois cette correspondance effectuée, on remplace la valeur du pixel dans l'image gauche par la distance entre les deux points.

Deux sortes de technique existent pour retrouver la carte de disparité, soit il s'agit de minimiser une énergie locale, généralement par un calcul de corrélation sur des ensembles de pixels ordonnés, appelés "patches", soit il s'agit de minimiser une énergie globale de tel sorte que $E(d) = E_{data}(d) + E_{smooth}(d)$.

Nous parlerons dans ce chapitre, dans un premier temps des méthodes dites locales par corrélation de patches, puis nous aborderons l'état de l'art des méthodes globales, enfin nous expliquerons la méthode choisie.

Méthodes locales par corrélations de patches

La fonction de corrélation peut être calculée soit entre pixels soit entre voisinages. Toutefois sur de simples pixels trop d'erreurs existent et notre choix s'est donc porté sur une fonction de coût entre patches.

L'idée naturelle est de penser que si les points sont semblables, alors leurs voisinages le sont aussi. Cette hypothèse est très forte et nous en montrerons les limites. Pour mesurer cette similarité, on peut étudier différentes fonctions qui font une comparaison entre les voisinages comme la fonction de corrélation croisée centrée normalisée (ZNCC) μ , ou la fonction de somme des différences carrées normalisées (ZNSSD) ρ telles que :

$$\mu(d) = \frac{\sum_i \sum_j (I_l(x_i, y_j) - \overline{I_l(x, y)})(I_r(x'_i, y'_j) - \overline{I_r(x', y')})}{\sqrt{(\sum_i \sum_j (I_l(x_i, y_j) - \overline{I_l(x, y)})^2 \sum_i \sum_j (I_r(x'_i, y'_j) - \overline{I_r(x', y')})^2)}} \quad (1.12)$$

$$\rho(d) = \frac{\sum_i \sum_j ((I_l(x_i, y_j) - \overline{I_l(x, y)}) - (I_r(x'_i, y'_j) - \overline{I_r(x', y')}))^2}{\sqrt{(\sum_i \sum_j (I_l(x_i, y_j) - \overline{I_l(x, y)})^2 \sum_i \sum_j (I_r(x'_i, y'_j) - \overline{I_r(x', y')})^2)}} \quad (1.13)$$

Avec d la disparité, I_l l'image gauche et I_r l'image droite, on traite le point m_l de coordonnées (x, y) et m_r de coordonnées (x', y') et $\overline{I_l(x, y)}$ la moyenne des intensités des pixels centrés autour de (x, y) . Les indices i et j permettent de parcourir le voisinage

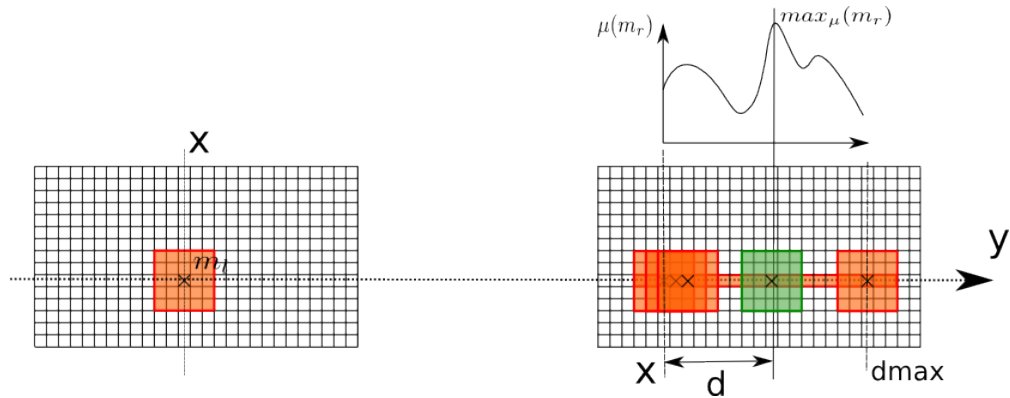
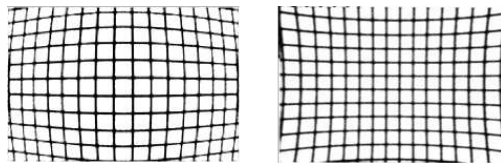


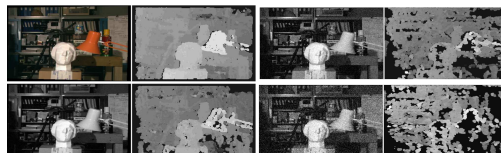
FIGURE 1.4.1 – Schéma de l'appariement, la fonction de corrélation est calculée le long de la même ligne de pixels, avec μ la fonction définie en 1.12, on trouve le pic de corrélation et le pixel associé à cette valeur maximale est choisi comme correspondant et d , la disparité, est calculée

Le problème dans la mise en correspondance est que l'hypothèse de voisinage semblable n'est pas vrai dans les cas exposés ci-dessous, ce qui rend cette étape finalement très complexe :

- des distorsions peuvent subsister dans les images, ces distorsions sont dues à la lentille (visible sur la figure 1.4) ;



- du bruit est présent dans les images (visible sur la figure 1.4) ;



- des surface peuvent être spéculaires, c'est-à-dire que la surface est brillante, le problème est que la position des reflets change avec la prise de vue (visible sur la figure 1.4) ;



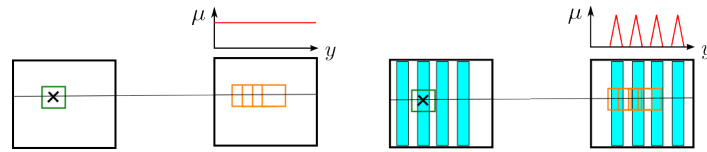
FIGURE 1.4.2 – Exemple de carte de disparité calculée avec une fenêtre 5x5 avec une similarité ZNCC



- de fortes transformations projectives peuvent réduire ou étirer certaines zones de l'image, ces problèmes sont dus à la manière de faire les prises de vues (visible sur la figure 1.4) ;



- les régions homogènes sont très difficiles à apparier car les pixels sont difficilement identifiables les uns des autres. L'appariteur qui cherche le patch le plus semblable ne sait pas lequel prendre, car il existe trop de pics de corrélation ; Le même problème apparaît avec des formes se répétant. Dans ce cas là, de trop nombreux pics de corrélations existent et il très difficile de faire confiance uniquement au maximum (visible sur la figure 1.4) ;



- des objets transparents ne peuvent pas être appariés. Ces cas sont anecdotiques ;
- de nombreuses occlusions apparaissent, ainsi un pixel de l'image gauche n'a pas automatiquement de correspondant dans l'image droite, ce qui est difficilement identifiable car la corrélation étant faite sur une ligne, un maximum de corrélation existe ;
- de même, des discontinuités peuvent être présentes. Ainsi au bord des objets, il est difficile de mettre en correspondance les pixels, ce qui a tendance à faire déborder les surfaces, c'est-à-dire que les surfaces sont agrandies de la moitié de la taille de la fenêtre de corrélation. Une surface sera donc de 2 pixels trop grande pour une fenêtre de corrélation 5×5 . Ce phénomène est appelé adhérence et est difficile à identifier comme le montre la figure 1.4 ;

Résoudre ces problèmes est difficile. Mais des solutions peuvent être apportées :

- Pour les distorsions, peu de solutions sont possibles, car elles dépendent de l'étape de calibrage ;
- Le bruit est à prendre en compte si le rapport signal sur bruit devient trop petit, ce qui est rarement le cas pour des images prises en extérieur avec de bonnes conditions d'éclairage ;
- Pour le problème des surfaces non lambertiennes, c'est-à-dire réfléchissantes ou brillantes, les correspondances ne sont pas bonnes et un simple seuillage sur les valeurs de corrélation permet de les éliminer comme mauvais appariement ;
- Aucune solution n'existe pour les fortes transformations projectives si la géométrie de la scène n'est pas parfaitement connue ;
- Les régions homogènes et les motifs se répétant sont des problèmes fréquents. Or les solutions ne sont pas simples. Mais les scènes extérieures ne contiennent que peu de régions homogènes. Ces scènes offrent toujours un motif différent qui permet à la corrélation de trouver un pic, souvent bien placé, contrairement à, par exemple, une scène intérieur d'un bâtiment comprenant de nombreuses surfaces homogènes et de motifs répétitifs. Les scènes extérieures étudiées n'en comporte pas ;
- Les scènes naturelles présentent souvent de nombreuses occlusions et désocclusions, correspondant aux phénomènes d'occultation et d'apparition d'éléments entre les deux scènes (une feuille vue sur une image qui est cachée sur l'autre ou qui apparaît sur la deuxième) ;

Ces problèmes viennent, la plupart du temps, du fait que le patch choisi est de taille et de forme fixe. En effet, choisir un patch de forme et de taille fixe ne permet pas de prendre en compte les problèmes de plans non fronto-parallèles et les problèmes de discontinuités de profondeur (c'est-à-dire que pour un pixel correspondant, les fonds des deux patches seront différents).

L'état de l'art nous apprend que plusieurs algorithmes existent pour modifier la forme

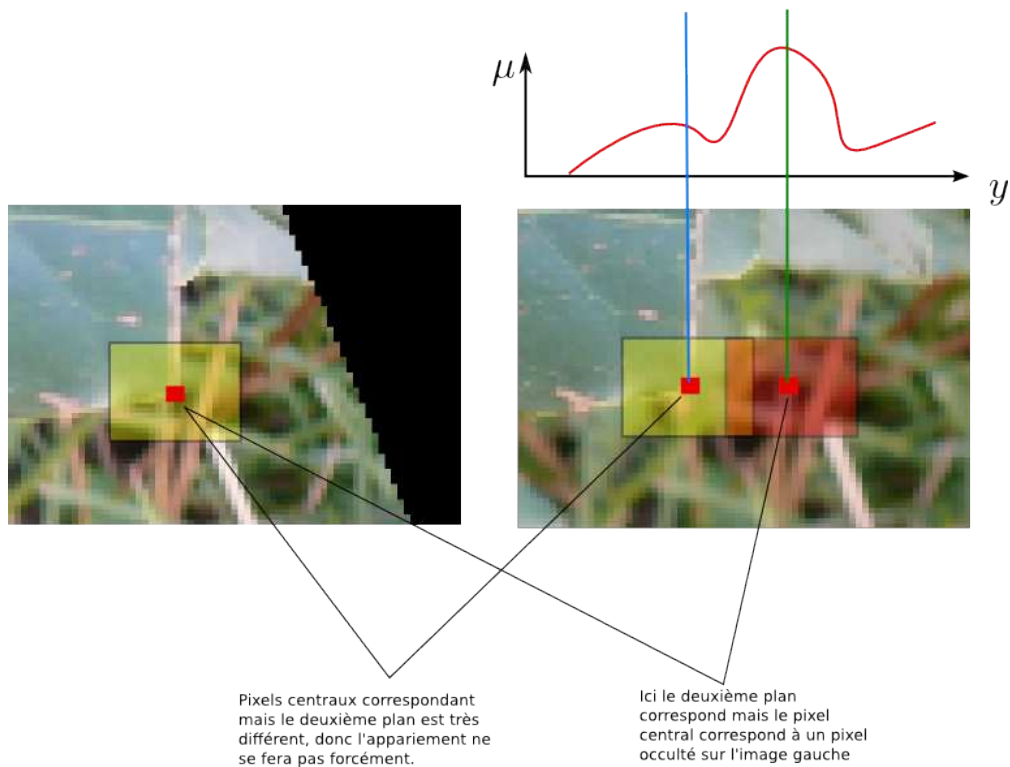


FIGURE 1.4.3 – Schéma du problème de patch à taille fixe. Ici, les patches jaunes qui sont censés correspondre n'ont pas le même fond et la majorité de la fenêtre appartient au fond. Ainsi le patch mis en correspondance sera celui en rouge et la disparité sera mauvaise. L'effet peut être inverse et c'est un pixel du fond qui aura la disparité de l'objet. Ce sont ces phénomènes qui provoquent les adhérences.

et la taille du patch dans le but de minimiser les problèmes expliqués ci-avant. Il existe différentes approches définies ci-dessous et résumées dans la figure 1.4 :

- les fenêtres décalables ou "shiftable windows" [Scharstein and Szeliski, 2002] qui n'imposent pas la position centrale du pixel traité qui peut ainsi se retrouver au bord du patch traité ;
- les fenêtres multiples ou "multiple windows" [Hirschmuller et al., 2002] qui se composent en plusieurs sous patches, qui sont gardés ou non, rendant le patch composé rarement carré ou rectangulaire ;
- les fenêtres variables ou "variable windows" [Veksler, 2003], qui se comportent un peu comme les "shiftable windows" mais dont la taille du patch n'est plus contrainte et peut ainsi s'agrandir ou se réduire ;
- les approches basées sur une segmentation de l'image [Gerrits and Bekaert, 2006] ;

Dans ces dernières approches la forme du patch dépendra, soit du nombre de pixels dans la zone de segmentation, soit d'une estimation d'un poids dépendant de la segmentation et de la distance colorimétrique [Tombari et al., 2007], ainsi les pixels

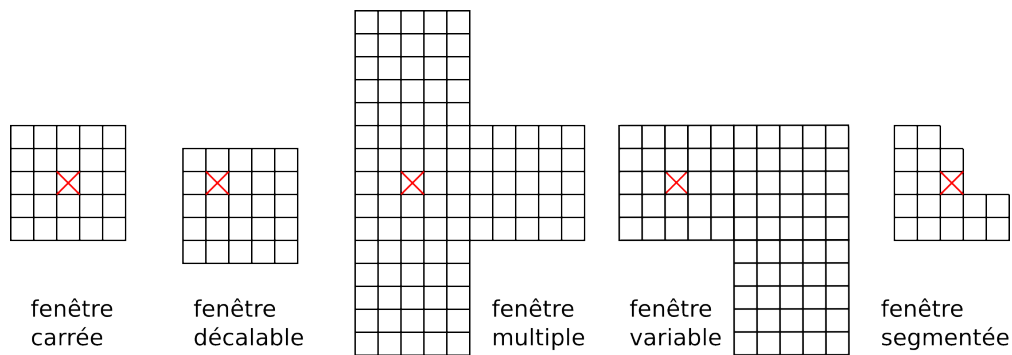


FIGURE 1.4.4 – Schéma des différentes solutions par modification de la forme des patches.

qui ne sont pas dans la même zone de segmentation auront un poids plus ou moins élevé en fonction de leur ressemblance colorimétrique.

D'autres approches sont basées sur un changement du noyau de calcul de corrélation. Usuellement un noyau uniforme est choisi, c'est-à-dire que chaque pixel a la même importance. D'autres noyaux, comme le noyau gaussien ou bilatéral, peuvent estimer un poids en donnant plus d'importance aux pixels les plus proches du pixel central [Yoon and Kweon, 2006].

Au vu des tests effectués, il est apparu que les approches basées sur les images de segmentations montraient de très bons résultats. En effet ces algorithmes offrent de bons résultats dans le cas d'images de galets ou de plantes, mais peuvent provoquer, si les différences de couleurs sont trop faibles, des phénomènes de sur-segmentation ou de sous-segmentation si les paramètres de segmentation ne sont pas choisis avec soin. Certains de ces algorithmes basés sur les images de segmentation considèrent les plans de segmentation comme non frontaux-parallèles et estiment une pente différente. Ces méthodes bien qu'efficaces dans le cas de surfaces planes fonctionnent mal dans le cas d'images de feuillages, la présence de trop nombreux plans empêche ces algorithmes d'être appliqués.

Ainsi pour garder une robustesse, en temps raisonnable (moins de 30 sec par couples d'images), nous avons choisi d'utiliser des méthodes locales par patches variables utilisant les images de segmentations. Ces algorithmes ont montré la plus grande robustesse pour nos images.

1.4.1 Méthodes globales

Il existe des techniques qui minimisent une fonction de coût global et permettent ainsi d'obtenir la carte de disparité directement. Dans ces techniques, différents algorithmes classés parmi les premiers dans l'évaluation de l'université de Middlesbury [Scharstein and Szeliski, 2002] utilisent des techniques de minimisations par "graph cut" [Kolmogorov and Zabih, 2001], ou par propagation de confiance "belief propaga-

tion" [Klaus et al., 2006].

L'initialisation de ces algorithmes utilise une disparité grossièrement trouvée par corrélation de patches. Cependant, les meilleurs résultats pour les cartes de disparité par minimisation d'énergie sont obtenus avec les meilleures initialisations. Or l'initialisation s'effectue au moyen d'une méthode locale par patch. Ainsi, pour obtenir la meilleure initialisation, les patches utilisés sont de formes variables et se servent d'images de segmentation. Enfin, ces algorithmes utilisent une optimisation finale par coopération, appelée "cooperative optimization" [Wang and Zheng, 2008] pour améliorer les résultats.

Malheureusement ces techniques ne fonctionnent que si le nombre de pixels n'est pas trop grand ou que l'espace de disparité n'est pas trop élevé (c'est-à-dire qu'il n'y pas beaucoup de différence de profondeur dans la scène). Or, nous travaillons avec de grandes images (>10millions de pixels) comprenant de nombreux niveaux de disparité (>200) rendant l'utilisation de ces techniques, appliquées directement, impossible.

Une possibilité serait alors de se servir d'images sous-échantillonnées, mais la perte d'information limiterait beaucoup l'intérêt de ces techniques. Une autre méthode serait d'utiliser un mosaïquage de l'image et d'utiliser les algorithmes sur des sous-images en vérifiant le maintien d'un recouvrement suffisant.

Mais la lenteur, les difficultés de mise en place et le manque de robustesse liés aux images prises sans contraintes nous ont fait préférer les méthodes locales.

1.4.2 Approche locale multi-échelles

Au vu de l'état de l'art nous avons choisi d'utiliser un algorithme robuste à patch fixe [Leloglu et al., 1998], qui utilise une approche hiérarchique. C'est-à-dire que la disparité est calculée sur une image sous-échantillonnée deux fois. Une fois la carte de disparité calculée sur les images sous-échantillonnées, une nouvelle carte est calculée en remontant la pyramide d'images, la disparité étant optimisée par relaxation.

Mais pour rendre cet algorithme plus robuste aux problèmes classiques liés aux patches fixes, nous avons choisi d'appliquer un algorithme de patch variable par segmentation. Cet algorithme hybride, utilisant des correspondances à patch variable de manière hiérarchique, a donné de très bons résultats.

Nous montrerons ici le fonctionnement de l'algorithme.

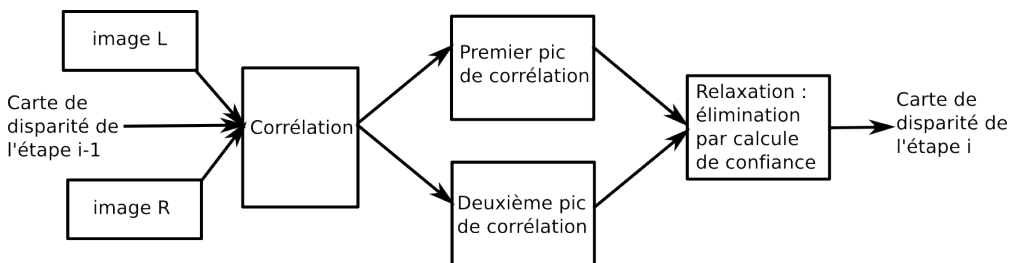


FIGURE 1.4.5 – Schéma de l'algorithme original par approche multi-échelle.

Simplement, sur la première image de la pyramide, c'est-à-dire la plus sous-

échantillonnée, une corrélation par patch est effectuée le long de la ligne épipolaire correspondante. La mesure de similarité utilisée entre les deux patches est une mesure de corrélation croisée centrée et normalisée (ZNCC) définie par l'équation 1.12. On obtient ainsi une première carte de disparité montrée à la figure 1.6(a).

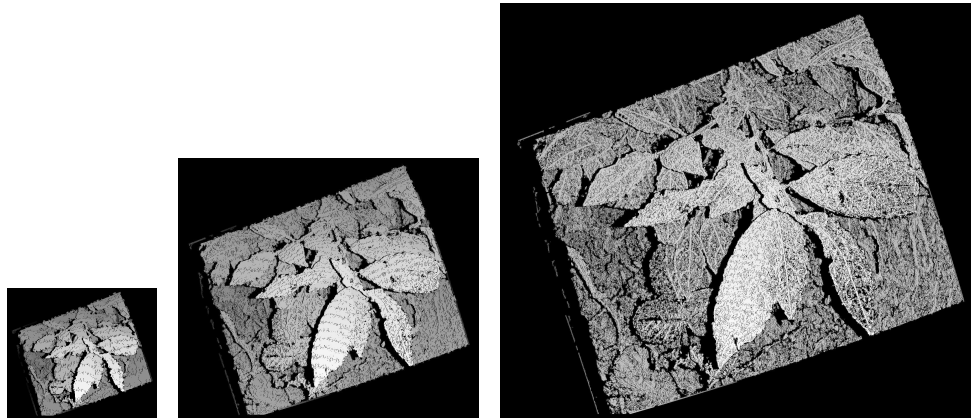


FIGURE 1.4.6 – Carte de disparité sur l'image rectifiée de feuilles aux différents rangs de la pyramide.

Nous avons choisi d'utiliser une image de segmentation permettant d'éviter les effets d'adhérence observés sur la figure 1.4.8, c'est-à-dire que la disparité semble adhérer aux surfaces et provoque une augmentation des surfaces de disparité, cette augmentation étant la moitié de la taille de la fenêtre de corrélation. Cette segmentation se fait de manière colorimétrique, en utilisant des algorithmes classiques tels que l'algorithme de mean-shift [Comaniciu and Meer, 2002], un algorithme de segmentation pyramidale [Lifshitz and Pizer, 1990], ou un algorithme de segmentation par partage et réunion de régions [Suk and Chung, 1983]. Le schéma de l'algorithme devient alors comme celui de la figure 1.4.2

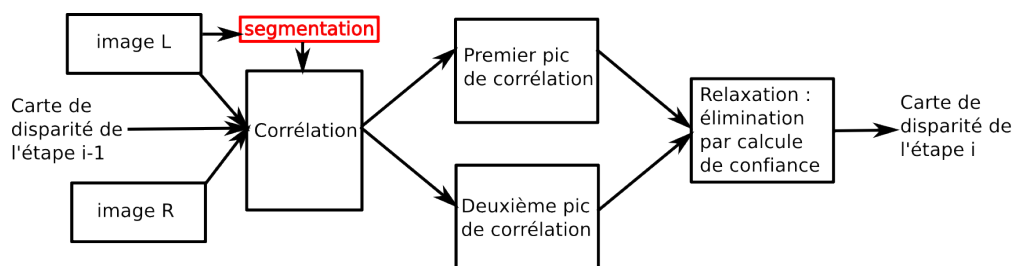


FIGURE 1.4.7 – Schéma de l'algorithme utilisé.

Une fois l'image de segmentation créée, la forme de la fenêtre de corrélation dépendra de la zone de segmentation à laquelle appartient le pixel traité et ses voisins. Le problème, soulevé par l'utilisation des images de segmentation, est que le nombre de paramètres déjà nombreux augmente encore. Dans le cas de mauvais paramètres sur

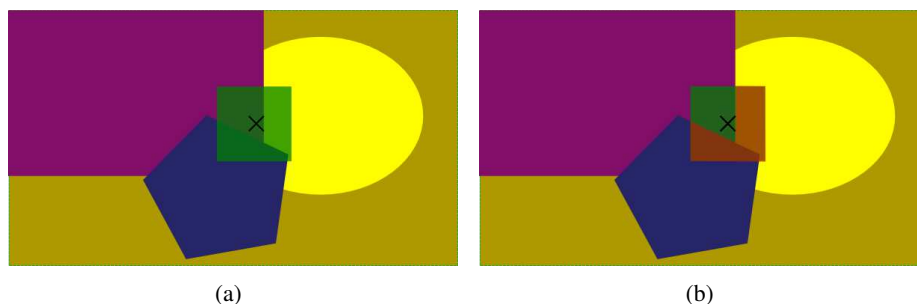


FIGURE 1.4.8 – Étapes de sélection de la forme de la fenêtre de corrélation, en (a) le pixel choisi appartenant à la zone violette avec le patch de forme carrée en vert et en (b) le même patch avec en vert les parties prises en compte et en rouges celles rejetées.

Algorithm 1 Relaxation :

```

while  $t_{max} > t_{min}$  do
  for  $i = 1$  à  $N$  do
     $\forall P_i$ , calcul de  $s_i$ 
    if  $s_i > t_{max}$  then
      L'appariement est accepté comme juste
      Rejet de tous les points ne respectant pas la contrainte d'ordre
    end if
    if  $s_i < t_{min}$  then
       $P_i$  supprimé
    end if
    augmentation de  $t_{min}$ 
    diminution de  $t_{max}$ 
  end for
end while

```

la segmentation, des phénomènes de sous-segmentation peuvent apparaître ramenant le problème au cas de mise en correspondance avec taille de patch fixe. Dans le cas de phénomènes de sur-segmentation, trop peu de pixels sont appariés. Ainsi, il faut bien choisir les paramètres de segmentation qui peuvent varier en fonction des images. Malgré ces problèmes, utiliser l'image de segmentation rend les algorithmes plus robustes et supprime la plupart des effets d'adhérences.

Une fois cette carte de disparité de niveau inférieur calculée, on se sert de celle-ci comme point de départ pour le calcul de la carte de niveau supérieur. Ainsi, pour chaque correspondant de l'image de niveau inférieur, les points inclus dans une région autour de ce point sont inclus à la recherche. Une fois les deux cartes de disparité calculées, correspondant aux deux candidats, c'est-à-dire aux deux premiers pics de corrélation, une étape de relaxation est effectuée entre les deux cartes permettant de choisir le bon match comme expliqué dans l'algorithme 1.



(a)



(b)

FIGURE 1.4.9 – Carte de disparité sur les images de galets avec et sans segmentation. Sans segmentation, (a) les parties grises dépassent, c'est-à-dire que la disparité a adhéré et que des pixels ont été mal appariés. Avec segmentation (b), les contours sont mieux respectés mais un plus grand nombre de pixels ne trouvent pas de correspondant.

On précisera que dans l'algorithme 1 on définit s_i tel que :

$$s_i = \sum_{k=1}^N c_k W(\|\overrightarrow{m_k m_i}\|)$$

et

$$W(\rho) = \frac{1}{1 + e^{b(\rho-r)}}$$

avec N le nombre de pixels, b un facteur de transition et r le rayon du voisinage.

On remonte ainsi la pyramide des valeurs de telle manière que les disparités de l'étape précédente servent à trouver le bon match, en respectant la contrainte d'ordre et en minimisant l'espace de recherche. Une fois les bons appariements calculés, les cartes de disparités peuvent être créées.

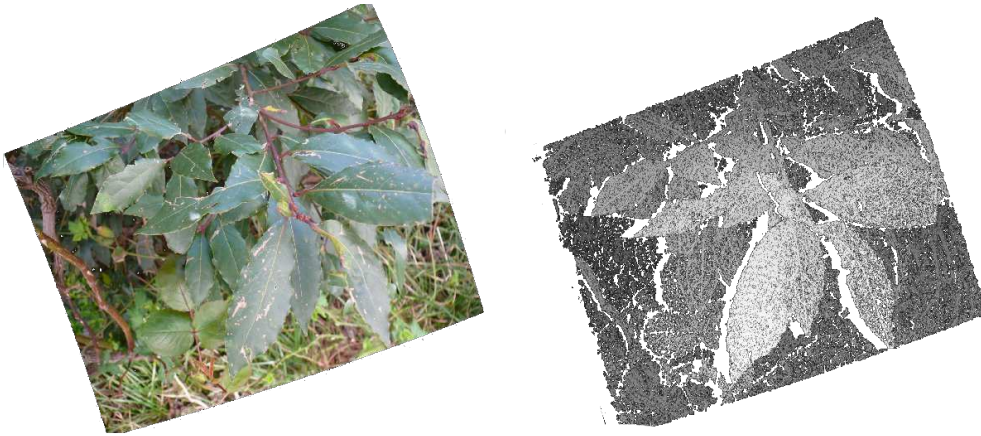


FIGURE 1.4.10 – Carte de disparité sur l'image rectifiée de feuilles.

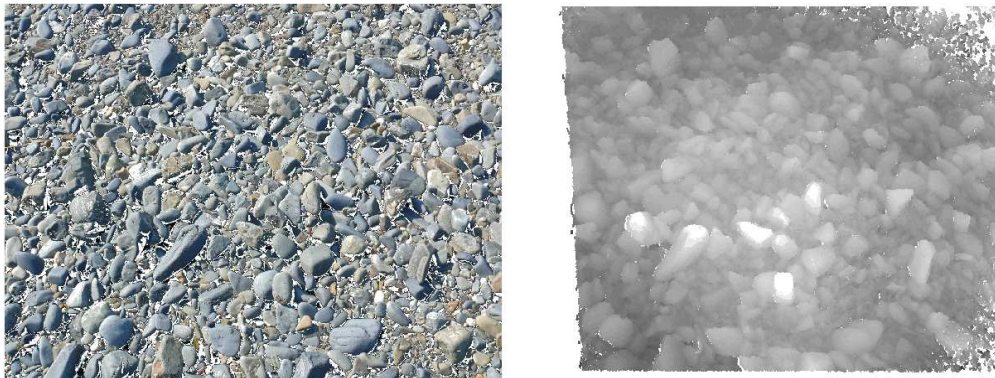


FIGURE 1.4.11 – Carte de disparité sur l'image originale de galets, l'erreur sur les coins est liée à une mauvaise estimation de la distorsion.

Il est intéressant de noter que l'utilisation d'algorithmes comme l'appariement de points invariants par échelles et la rectification épipolaire par minimisation polaire permet de calculer les disparités même entre des photos prises à des échelles différentes comme on peut le voir sur la figure 1.4.12.

La carte de disparité peut, une fois triangulée, présenter un effet de "marche d'escalier" ou "stepping", visible dans le nuage 3D de la figure 1.4.2. Cet effet est

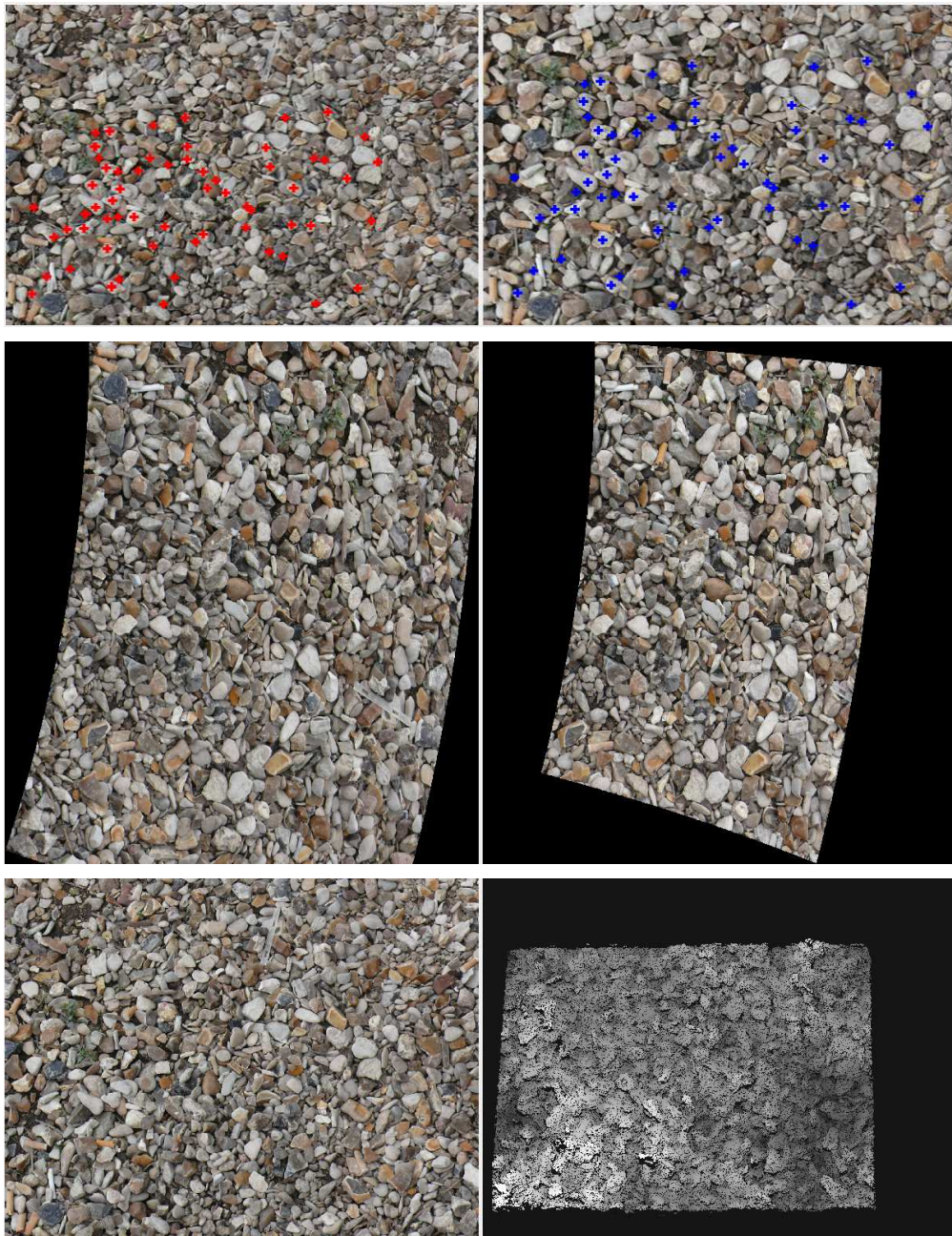


FIGURE 1.4.12 – De l'appariement à la carte de disparité d'un couple d'images de graviers pris à des échelles différentes

lié au fait que la disparité est une valeur discrète. Pour supprimer cet effet, une étape de corrélation sub-pixellique est effectuée.

Deux approches existent, la première estime une parabole à partir des valeurs de corrélation et prend le maximum de celle-ci comme expliqué sur la figure 1.4.14. La deuxième effectue la corrélation sur une image sur-échantillonnée comme expliqué

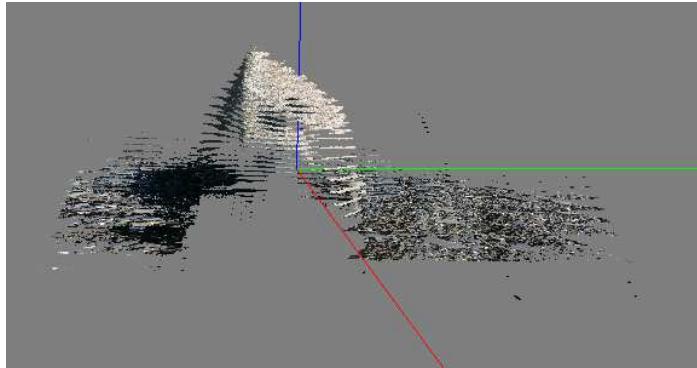


FIGURE 1.4.13 – Résultat d'un nuage de point 3D, on aperçoit ce phénomène de marche d'escalier lié aux valeurs discrètes de la disparité

sur la figure 1.4.15.

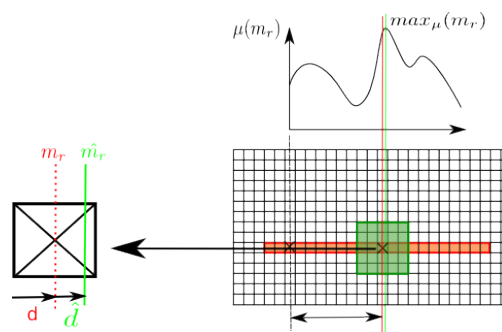


FIGURE 1.4.14 – Schéma de l'algorithme de corrélation sub-pixellique par estimation de la parabole.

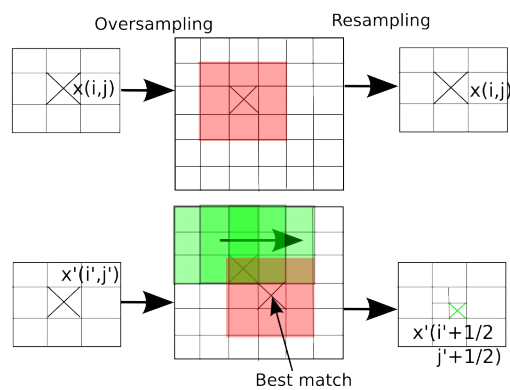


FIGURE 1.4.15 – Schéma de l'algorithme de corrélation sub-pixellique avec un facteur de sur-échantillonnage de 2.

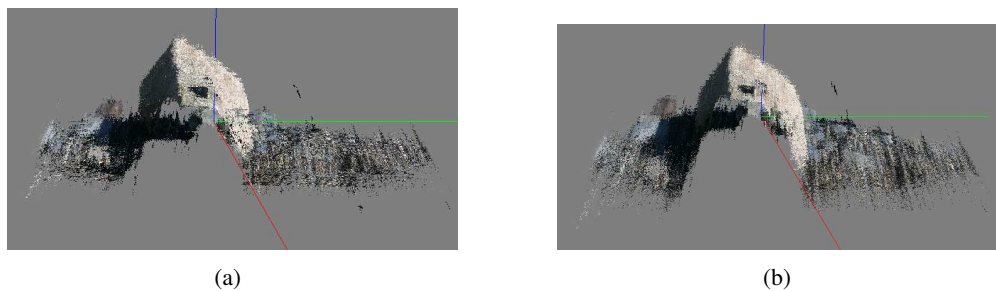


FIGURE 1.4.16 – (a) résultat du nuage triangulé avec le maximum d’une parabole, (b) résultat du nuage triangulé avec l’image sur-échantillonnée

On peut voir sur la figure 1.4.2 que le meilleur rapport qualité/temps a été obtenu avec l’estimation du maximum de la parabole. Nous avons donc choisi pour notre chaîne de reconstruction d’utiliser cette approche.

1.5 Auto-calibrage

Le calibrage permet de connaître l’ensemble des paramètres décrivant l’espace projectif, c’est-à-dire la correspondance entre le monde réel et l’image. Ces paramètres se subdivisent en deux parties distinctes : les paramètres intrinsèques correspondant à l’état de la caméra et les paramètres extrinsèques correspondant au passage de l’espace Euclidien réel à sa projection dans l’espace image. Or il est nécessaire de connaître le calibrage si des mesures doivent être effectuées car les photos, placées dans un espace projectif comme sur la figure 1.5.1, ne respectent pas les angles, les parallélismes et encore moins les longueurs.

Deux possibilités s’offrent pour trouver ces paramètres, soit un calibrage manuel, soit un calibrage automatique. Dans le premier cas, un objet référence est pris sous différents angles et, en mettant en correspondance la connaissance sur l’objet et sa projection sur l’image, on trouve une solution pour les matrices de paramètres intrinsèques et extrinsèques. Dans le cas de l’auto-calibrage cette étape est facultative, c’est directement sur les objets de la scène que seront effectuées les correspondances. Ainsi, par estimation des correspondances dans les différentes images et sans rien connaître de la scène prise en photo, une solution sera trouvée. Dans l’optique de permettre des mesures à partir de photos prises sans contrainte, nous nous sommes limités à l’étude du cas de l’auto-calibrage, même si dans la plupart des cas ce calibrage est moins fiable que le calibrage manuel.

Généralement l’auto-calibrage se fait en trois étapes principales :

- la reconstruction projective par calcul de correspondances sur un couple d’images ;
- les autres images sont ajoutées une par une au système calibré par une transformation linéaire directe ;
- un raffinement global effectué par ajustement de faisceaux ;

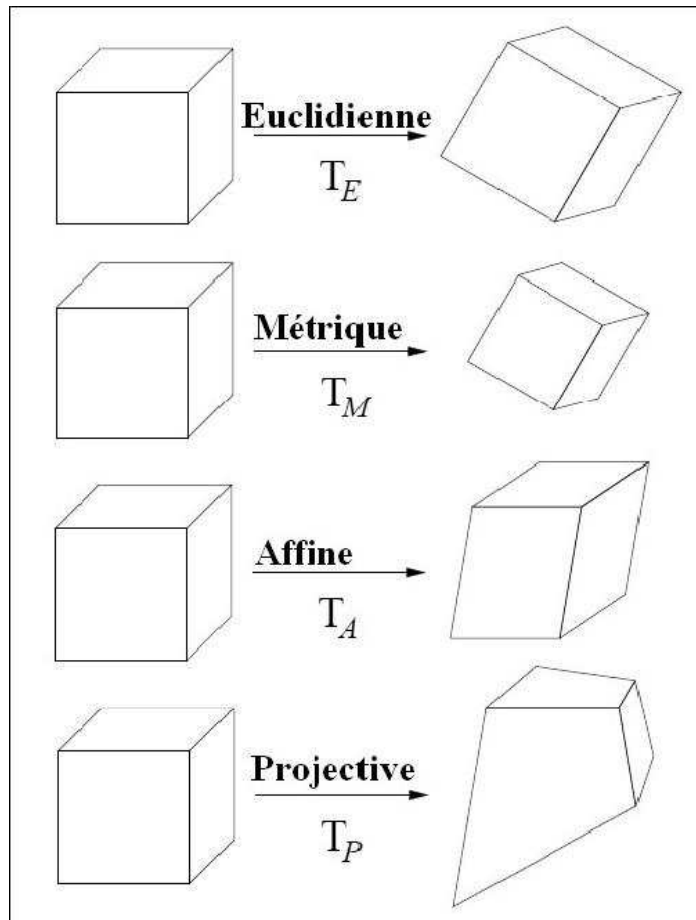


FIGURE 1.5.1 – Schéma des différentes transformations d’espaces ; euclidien : tout est conservé ; métrique : l’échelle n’est plus conservée ; affine : les angles et l’échelle ne sont plus conservés ; projectif : l’échelle, les angles et le parallélisme ne sont plus conservés.

Les techniques basées sur ces trois étapes comme [Pollefeys et al., 1999], [Pollefeys et al.,], [Faugeras et al., 1992], [Triggs, 1997] ou [Snavely et al., 2008] ont montré de très bons résultats.

Ainsi, l’auto-calibrage utilisé cherchera des correspondances robustes dans les images comme montré à la section précédente (SIFT par exemple, SURF dans notre cas). Puis, on estimera les paramètres projectifs pour une seule paire de caméras en utilisant la résolution de la matrice essentielle, matrice expliquant le déplacement rigide entre les deux caméras, par un amélioration du "Nistér 5-point solution" [Stewenius et al., 2006]. Enfin, à partir de cette paire initiale astucieusement choisie et de la matrice essentielle les caméras sont rajoutées une par une. La position de la nouvelle caméra relativement aux autres sera estimée. Cette estimation sera effectuée par un raffinement global par ajustement de faisceaux, c’est-à-dire qu’à chaque nouvelle image, toutes les positions des caméras seront ré-estimées. Cette

technique est basée sur les travaux de [Triggs et al., 2000] et plus récemment de [Lourakis and Argyros, 2004]. Triggs et al [Triggs et al., 2000] avec l'utilisation des techniques de matrices creuses [Szeliski and Kang, 1994] ont démontré la fiabilité et la robustesse de l'ajustement de faisceaux et ont permis la popularisation de leur utilisation pour l'estimation du calibrage global. Des travaux plus récents, [Brown and Lowe, 2005] et [Snavely et al., 2008] ont montré que l'ajustement de faisceaux était particulièrement robuste, et pouvait compenser certaines imprécisions de l'auto-calibrage.

1.5.1 Auto-calibrage d'un couple d'image

Généralement, l'auto-calibrage d'un couple d'image se fait par l'utilisation des contraintes sur la conique absolue et celles des équations de Kruppa [Pollefeys and Van Gool, 1997]. Une conique absolue est une conique imaginaire dans le plan à l'infini, dont la particularité est d'avoir toujours la même projection sur l'image si les paramètres intrinsèques sont constants. La projection dans l'image d'une conique absolue est invariante par rotation et translation. En utilisant la définition du dual de la conique absolue, il est possible de définir un certain nombre de contraintes. De plus, les équations de Kruppa [Kruppa, 1913] imposent que les projections des coniques absolues satisfassent les contraintes épipolaires.

Ces contraintes serviront pour le calcul des matrices de projection (du projectif au métrique) d'un couple d'images astucieusement choisies [Pollefeys et al.,], [Faugeras et al., 1992]. Mais récemment, de nouvelles techniques ont vu le jour pour rendre plus robuste le passage du projectif au métrique [Nistér, 2004]. Le premier couple sera choisi comme celui possédant le plus de points appariés en commun avec les autres caméras.

Contraintes algébriques

Une fois le premier couple trouvé, il faut trouver le déplacement entre ces deux images. Or, pour cela, il est nécessaire de connaître la matrice essentielle qui se déduira d'un certain nombre de contraintes algébriques. Nous présenterons dans cette partie les contraintes utilisées pour l'obtention de cette matrice E , composée de 5 paramètres, qui décrit le déplacement rigide entre deux positions de caméras. La première contrainte est celle de coplanarité présentée par Longuet-Higgins en 1981 dans "nature" [Longuet-Higgins, 1981], appelée contrainte épipolaire, et résumée par la matrice fondamentale F :

$$m_r^T F m_l = 0 \quad (1.14)$$

Or, selon [Faugeras et al., 1992], F est de rang 2, on en déduit une deuxième contrainte :

$$\det(F) = 0 \quad (1.15)$$

La matrice fondamentale peut être estimée sans connaître les paramètres de calibrage, uniquement par la connaissance de correspondances entre les deux images. Toutefois, cette matrice contient toutes les informations permettant de passer d'une caméra

à l'autre. Soit, les matrices de paramètres intrinsèques K_l et K_r correspondent aux paramètres internes des deux images. Les paramètres extrinsèques R et t , respectivement la matrice de rotation et le vecteur translation, correspondent au déplacement rigide entre l'image gauche et l'image droite. On peut alors exprimer la matrice F telle que :

$$F \equiv K_r^{-T} [t]_x R K_l^{-1} \quad (1.16)$$

Ainsi, en connaissant les matrices K_l et K_r les matrices de paramètres intrinsèques des deux caméras connues grâce aux EXIF contenus dans chaque image, et en les pré-multipliant aux points appariés m_l et m_r qui donneront les points μ_l et μ_r , on spécialise l'équation 1.14 :

$$\mu_r^T E \mu_l = 0 \quad (1.17)$$

Il existe ainsi une relation entre la matrice essentielle E et F :

$$F = K_r^{-T} E K_l^{-1} = K_r^{-T} [t]_x R K_l^{-1} \quad (1.18)$$

On en déduit que $E = [t]_x R$. Il faut donc trouver un moyen d'estimer la matrice essentielle, car elle contient les informations sur le déplacement rigide entre les deux caméras.

Pour calculer cette matrice, on dispose des contraintes sur les coniques absolues, or il est dit dans [Pollefeys and Van Gool, 1997] que les deux valeurs singulières non-nulles de E sont égales, ce qui se traduit par l'équation :

$$E E^T E - \frac{1}{2} \text{trace}(E E^T) E = 0 \quad (1.19)$$

Les contraintes 1.17, 1.15 et 1.19 seront utilisées ci-dessous pour estimer la matrice essentielle permettant de retrouver t et R décrivant le déplacement rigide entre les deux caméras.

Estimation du déplacement rigide entre le premier couple d'images

De nombreux algorithmes existent dont le "8-points" [Longuet-Higgins, 1981] et "7-points" [Hartley and Zisserman, 2004] et permettent d'estimer E à partir de couples de points appariés. Or, la plupart de ces algorithmes ne sont pas robustes, ainsi les études ont montré que l'algorithme "5-points" [Nistér, 2004] et sa version modifiée [Stewenius et al., 2006] donnent les meilleures fiabilités et ainsi ont été choisies. La résolution du "5-point relative pose problem" de Nistér estimera la position entre le premier couple de photos et servira d'auto-calibrage pour celui-ci. L'algorithme comporte plusieurs phases :

- Une réécriture de l'équation 1.17 pour chacun des 5 points appariés telle que

$$\tilde{m} \tilde{E} = 0 \quad (1.20)$$

où

$$\tilde{m} \equiv [m_{l1} m_{r1} \ m_{l2} m_{r1} \ m_{l3} m_{r1} \ m_{l1} m_{r2} \ m_{l2} m_{r2} \ m_{l3} m_{r2} \ m_{l1} m_{r3} \ m_{l2} m_{r3} \ m_{l3} m_{r3}] \quad (1.21)$$

et \tilde{E} correspond à la vectorisation de la matrice E ligne par ligne. En accolant les 5 couples de points \tilde{m} on obtient une matrice 5×9 . On cherche alors la solution de cette équation linéaire, soit par décomposition en valeurs singulières soit, et c'est le cas ici, par décomposition QR. On trouve alors 4 matrices 3×3 X, Y, Z, W telles que

$$E = xX + yY + zZ + wW \quad (1.22)$$

avec x, y, z, w des scalaires définis à un facteur d'échelle près, en définissant généralement $w = 1$. Dans un cas bien déterminé, les 4 vecteurs singuliers X, Y, Z, W correspondent aux 4 valeurs singulières les plus basses. Sinon, d'autres étapes doivent être effectuées.

- On insère l'équation précédente 1.22 dans les contraintes cubiques (1.15 et 1.19) et on obtient alors une matrice A de dimension 10×20 après avoir opéré une élimination à pivot-partiel de Gauss-Jordan.
- On définit trois nouvelles équations en z comme combinaisons linéaires des précédentes et que l'on arrange dans une matrice B de taille 3×3 contenant les polynômes en z . En sachant que $[x \ y \ 1]^T$ est un vecteur solution de B , le déterminant est donc un polynôme de degré 10, que l'on appellera $\langle p \rangle$.
- On cherche la racine de $\langle p \rangle$ en utilisant une séquence de Sturm [Gellert et al., 1975]. Ainsi pour chaque racine de z ; x et y sont trouvés en utilisant le système d'équations B . La matrice essentielle est alors obtenue en remplaçant $[x, y, z, 1]$ dans l'équation 1.22
- Enfin on retrouve R et t par une décomposition en valeurs singulières de la matrice essentielle [Hartley and Zisserman, 2004]. En effet, en définissant la SVD de E tel que :

$$E \simeq USV^T \quad (1.23)$$

Avec $\det(U) > 0$ et $\det(V) > 0$.

On trouve alors t et R en se servant des théorèmes définis dans [Hartley and Zisserman, 2004] :

$$\begin{aligned} t &\simeq t_u \equiv [u_{13} \ u_{23} \ u_{33}]^T \\ R &= R_\alpha \equiv UDV^T \\ \text{ou } R &= R_\beta \equiv UD^T V^T \end{aligned}$$

avec

$$D \equiv \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ainsi la position du premier couple d'image l'une par rapport l'autre est connue. On aura donc la possibilité de trianguler les points saillants appariés dans un espace métrique. Ces points 3D retrouvés permettront ensuite une minimisation globale par ajustement de faisceaux.

1.5.2 Ajustement de faisceaux

Une fois le premier couple estimé, les caméras sont ajoutées l'une après l'autre en optimisant globalement la position des caméras par reprojections des points 3D dans les images. Chaque caméra est rajoutée, trouvant une initialisation des paramètres extrinsèques par une transformation linéaire directe (DLT) [Hartley and Zisserman, 2004]. Puis on trouve les meilleurs paramètres en appliquant une minimisation globale par ajustement de faisceaux.

L'algorithme de l'ajustement de faisceaux tente de minimiser un critère tel que :

$$\min_{P_j, X_i} \sum_{i=1}^n \sum_{j=1}^m d(P_j \cdot X_i, x_{i,j})^2 \quad (1.24)$$

avec P_j la $j^{\text{ème}}$ matrice de projection, X_i le $i^{\text{ème}}$ point 3D et $x_{i,j}$ le point correspondant sur l'image, où $P \cdot X$ le projeté du point 3D dans l'image et $d(.,.)$ la distance euclidienne.

Ainsi, l'algorithme se sert des appariements précédemment trouvés et de la reconstruction 3D partielle trouvée à l'étape précédente. En effet, les points 3D de l'étape précédente correspondent à des points appariés par SURF, ainsi quand on rajoute une nouvelle image, on cherche la position de l'image qui permet le mieux de mettre en correspondance, les points 3D projetés et leurs positions connues sur l'image.

Pour minimiser ce critère, un algorithme d'optimisation comme Levenberg-Marquardt itératif est utilisé [Mor, 1977]. Récemment une implémentation efficace, basée sur les travaux de [Lourakis and Argyros, 2004], et distribuée sous licence GPL a été développée pour une collection de photographies [Snavely et al., 2008]. C'est l'efficacité, la robustesse et la rapidité de cette implémentation ainsi que sa facilité de mise en place qui nous ont amenés à utiliser cette technique.

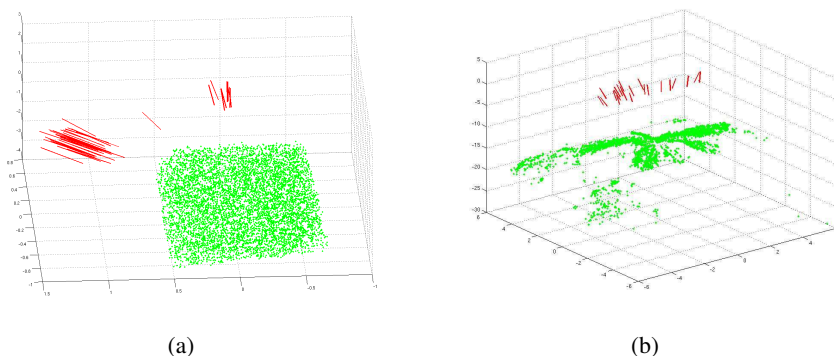


FIGURE 1.5.2 – Résultats de l'auto-calibrage sur les photos : (a) des galets et (b) des feuilles, avec en rouge la position des caméras et leurs orientations et en vert le nuage de points d'intérêt mis en géométrie métrique après l'étape de minimisation.

Discutions

Les résultats obtenus avec l'auto-calibrage ont, dans la plupart des cas, été satisfaisants. Malgré tout, nous nous sommes aperçus qu'un certain nombre de configurations de prises de vues posaient des problèmes à l'algorithme. Ces cas sont très spécifiques, il s'agit de la prise de "bandes" de photos avec un recouvrement de moins de 90%. En effet, cet algorithme offre les meilleures performances pour un maximum de recouvrement et des positions de prises de vues très variées. Ainsi, dans ce cas de figure de prise par "bande", il apparaît que des erreurs d'orientation des caméras se propagent à partir du premier couple comme visible dans la figure 1.5.3 montrant la reconstruction d'un mur.

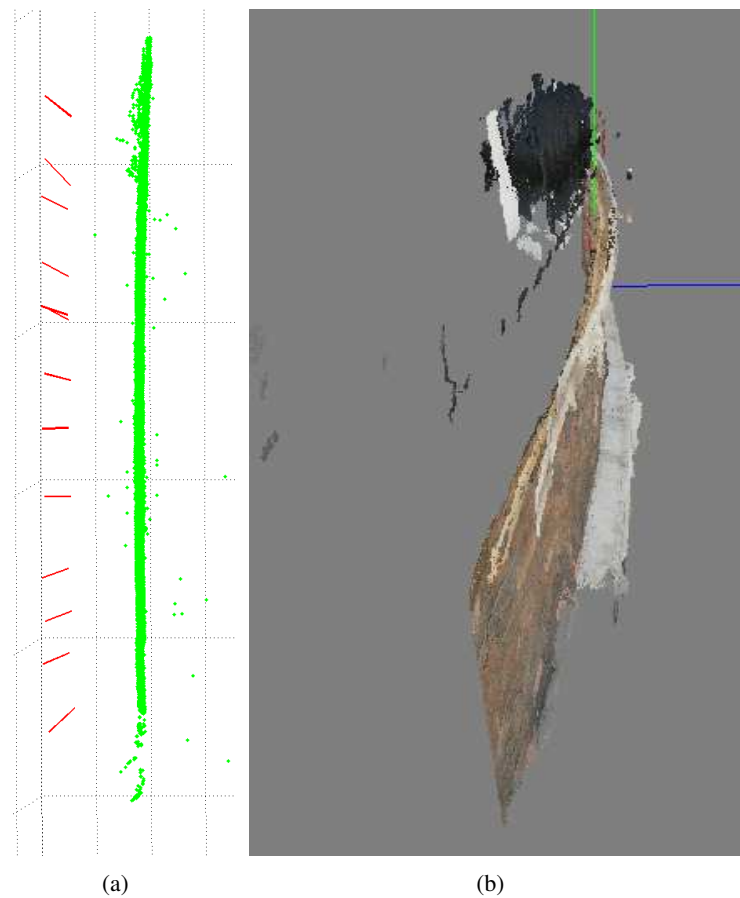


FIGURE 1.5.3 – (a) résultat du calibrage avec en rouge l'orientation des appareils photos et en vert une reconstruction clairsemée et en (b) la reconstruction finale obtenue sur laquelle on observe un "arrondissement" du mur qui est incohérent avec la réalité.

Ces problèmes sont inhérents au choix algorithmique, car comme toutes les images sont calibrées par rapport au premier couple, moins les images ont en commun avec ce premier couple, plus les erreurs peuvent se propager. Pour palier à ce problème, différentes stratégies sont envisageables, la première étant

de maximiser le nombre de photos. Une autre idée serait de prendre plusieurs occurrences de cette bande ou de modifier les angles de prises de vues. Mais toutes ces approches nécessitent une augmentation significative du nombre de photos.

Une autre possibilité est de fragmenter le calibrage en plusieurs étapes et de faire un ajustement de faisceaux sur les différentes reconstructions obtenues.

Une autre idée serait de contraindre la reconstruction avec un certain nombre de points cibles comme expliqué dans [Barazzetti et al., 2010].

Un manque de temps ne nous a pas permis d'explorer ces deux dernières pistes, nous nous sommes contenté d'augmenter le nombre de photos.

1.6 Triangulation et résultats

Une fois la géométrie des caméras connue et les appariements obtenus, il suffit alors de trouver le point 3D correspondant au couple de points dans l'espace métrique. Le problème est que les paramètres de calibrage ne sont pas parfaits. Or, dans un cas parfait, il existe un point se projetant parfaitement dans le couple de pixels, en sachant que ce point doit passer automatiquement par le centre optique de chaque image. Dans le cas réel, ce point n'existe pas, on cherche donc un compromis pour trouver le point approchant le plus proche de ce point idéal comme montré sur la figure 1.6.1

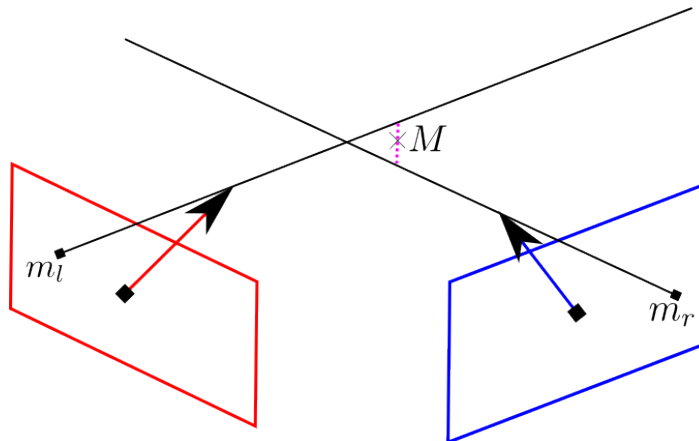


FIGURE 1.6.1 – Schéma du problème de la triangulation.

L'état de l'art, [Hartley and Sturm, 1997], nous dit qu'on peut estimer la position du point 3D par différents algorithmes simples, comme un algorithme linéaire utilisant une estimation de la position 3D par décomposition en valeurs singulières. Ainsi, un point 3D $X = (x, y, z, 1)$ est projeté dans l'image I_l en $\mathbf{u} = w(u, v, 1)$ par une matrice de projection P_l et dans I_r en $\mathbf{u}' = w'(u', v', 1)$ par la matrice de projection P_r avec P_l et P_r des matrices de dimensions 3×4 .

$$\mathbf{u} = P_1.X$$

$$\mathbf{u}' = P_2.X$$

Notons $p_{l,i}$ la i ème ligne de P_l et $p_{r,i}$ la i ème ligne de P_r . Ainsi :

$$\begin{aligned} wu &= p_{1,1}X, wv = p_{1,2}X, w = p_{1,3}X \\ w'u' &= p_{2,2}X, w'v' = p_{2,2}X, w' = p_{2,3}X \end{aligned}$$

En remplaçant w et w' on obtient :

$$\begin{aligned} up_{1,3}X &= p_{1,1}X \\ vp_{1,3}X &= p_{1,2}X \\ u'p_{2,3}X &= p_{2,1}X \\ v'p_{2,3}X &= p_{2,2}X \end{aligned}$$

Qui peut être réécrit :

$$AX = 0$$

Il reste ainsi à résoudre ce système. Une décomposition en valeurs singulières de A , matrice 4×4 , permet de trouver une valeur approchée de X .

1.6.1 Résultats et discussions

Une fois la triangulation effectuée pour tous les couples d'images successives, nous obtenons alors un ensemble de nuages dans le même espace. Toutefois en raison de nombreuses estimations et d'erreurs, que ce soit sur le calibrage ou sur la mise en correspondance, les nuages ne se superposent pas parfaitement et possèdent des points mal placés, appelés "outliers". Ainsi comme nous pouvons le voir sur les nuages de points ci-dessous, une étape de filtrage ainsi qu'une étape de suppression des redondances semblent nécessaires pour exploiter les nuages et permettre des mesures satisfaisantes.

Avec les trois exemples de résultats présentés en 1.6.2, 1.6.3 et 1.6.4, nous pouvons noter que de nombreux points indésirables restent présents. De plus, on s'aperçoit que les points correspondants aux plus petites disparités (c'est-à-dire les plus éloignés) sont imprécis dans leur position 3D. Une fois les nuages construits et mis en commun, de nombreux problèmes de superposition vont nous obliger à effectuer un traitement sur les nuages 3D.

Pour la végétation, des erreurs liées aux mouvements des feuilles provoquent un déplacement entre les différentes prises de vue. Ce mouvement change la disparité et par voie de conséquence, une fois triangulées, provoque les erreurs observées sur la figure 1.6.3. Quant aux erreurs présentes dans les autres images, il est difficile de les quantifier. C'est pour cela qu'une étape de filtrage et de fusion des nuages est nécessaire, nous expliquerons les différentes techniques utilisées dans le chapitre 4.

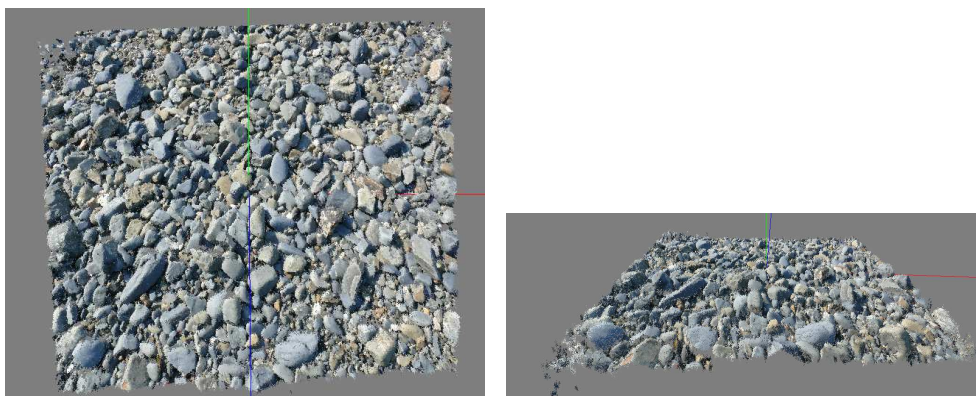


FIGURE 1.6.2 – Différentes vues d'un nuage 3D brut de galets obtenus à partir de 35 vues. Le faible nombre de niveau de disparité et la prise de vue quasi fronto-parallèle entraînent une bonne reconstruction possédant peu d'outliers.

conclusion

Nous avons vu dans ce chapitre tout les algorithmes nous permettant de passer d'une information image à une information 3D en respectant les différentes contraintes. Une fois ce nuage de point 3D obtenu, nous pouvons voir qu'il présente de nombreux défauts empêchant l'extraction de mesures fiables. Nous avons listé trois types d'erreurs : des points dont la position est aberrante, le fait que trop de points représentent la même information, erreur inhérente à la stéréo-vision pair par pair, et la présence de bruit sur les surfaces.

La chaîne présentée comporte un certain nombre de points critiques sans lesquels aucun résultat n'est possible. Le plus important est l'auto-calibrage. Le nombre important d'expériences pratiquées sur les algorithmes d'auto-calibrage nous ont permis de voir à quel point cette étape est cruciale. Quelques contraintes supplémentaires ont donc été imposées pour contourner les problèmes résiduels de cette étape qu'un travail plus approfondi aurait vraisemblablement permis de fixer.

Malgré tout, la chaîne de reconstruction 3D présentée nous est apparue comme la plus robuste aux contraintes. Il est donc impossible de résoudre ces problèmes directement en modifiant les algorithmes présentés sans nuire au respect des contraintes. Il nous faut donc passer par une étape de traitement du nuage de points 3D. Malheureusement, peu de recherches ont été effectuées pour les nuages créés de cette manière. En effet, si certaines techniques récentes offrent une meilleure reconstruction, elles sont inapplicables dans notre cas. Nous montrerons ainsi les stratégies mises en place pour palier à ces difficultés.

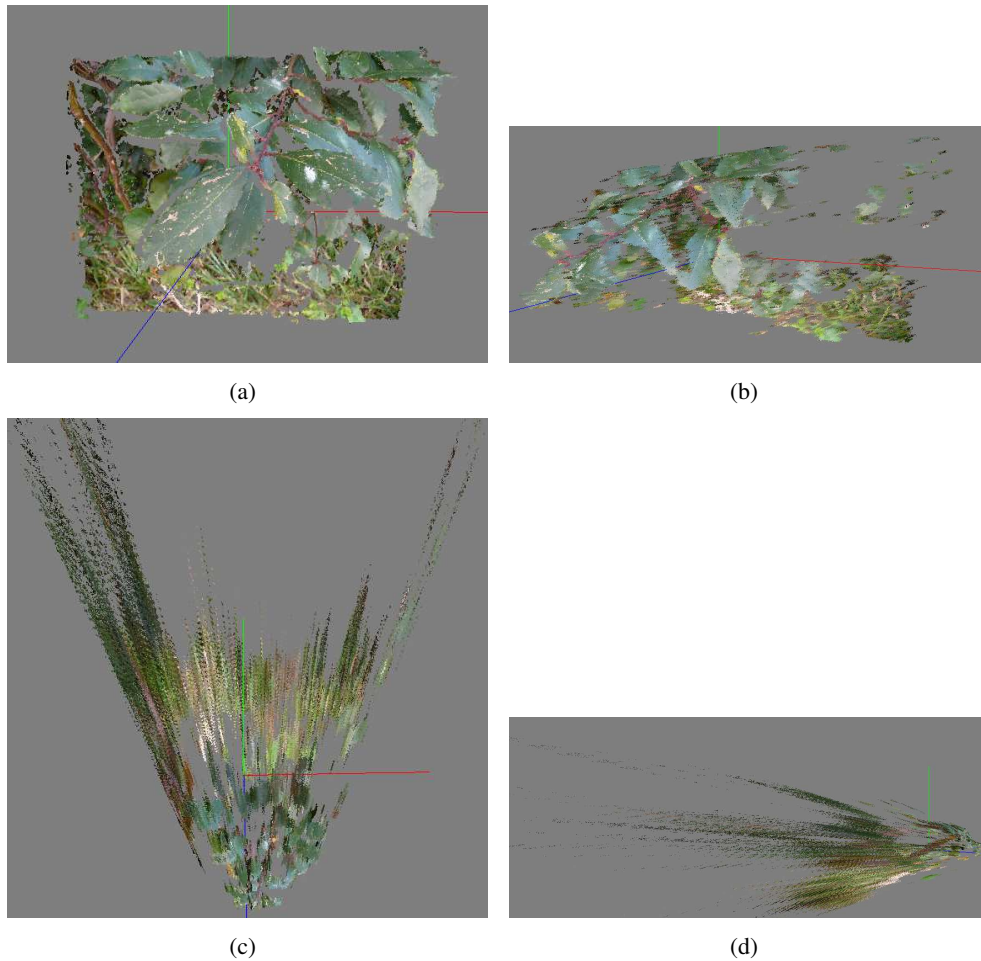
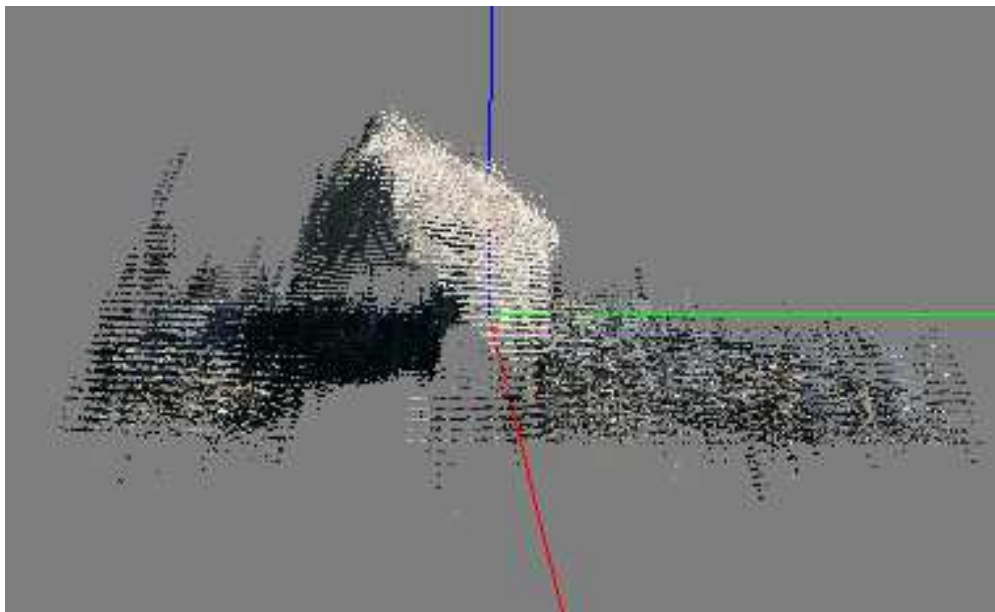


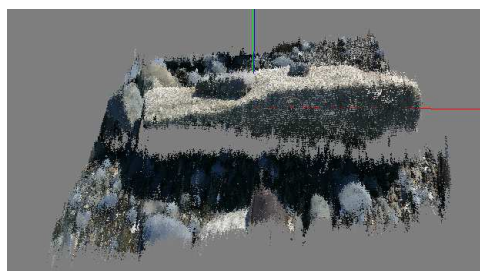
FIGURE 1.6.3 – Différentes vues d'un nuage 3D brut de feuilles obtenus à partir de 22 vues. Nous pouvons remarquer que les points les plus lointains vus sur les images (c) et (d) ont une disparité peu précise et ne permettent pas une interprétation satisfaisante des mesures que l'on pourrait effectuer. Ces erreurs proviennent de l'imprécision des points éloignés des caméras, ce qui entraîne ce phénomène "queue de comète" récurrent. Nous pouvons aussi remarquer que certains points semblent incohérents et qu'un effet de stepping persiste malgré un double sur-échantillonnage.



(a)



(b)



(c)

FIGURE 1.6.4 – Différentes vues d'un nuage 3D brut de galets obtenus à partir de 19 vues. En (a) l'effet de "stepping" est très visible.

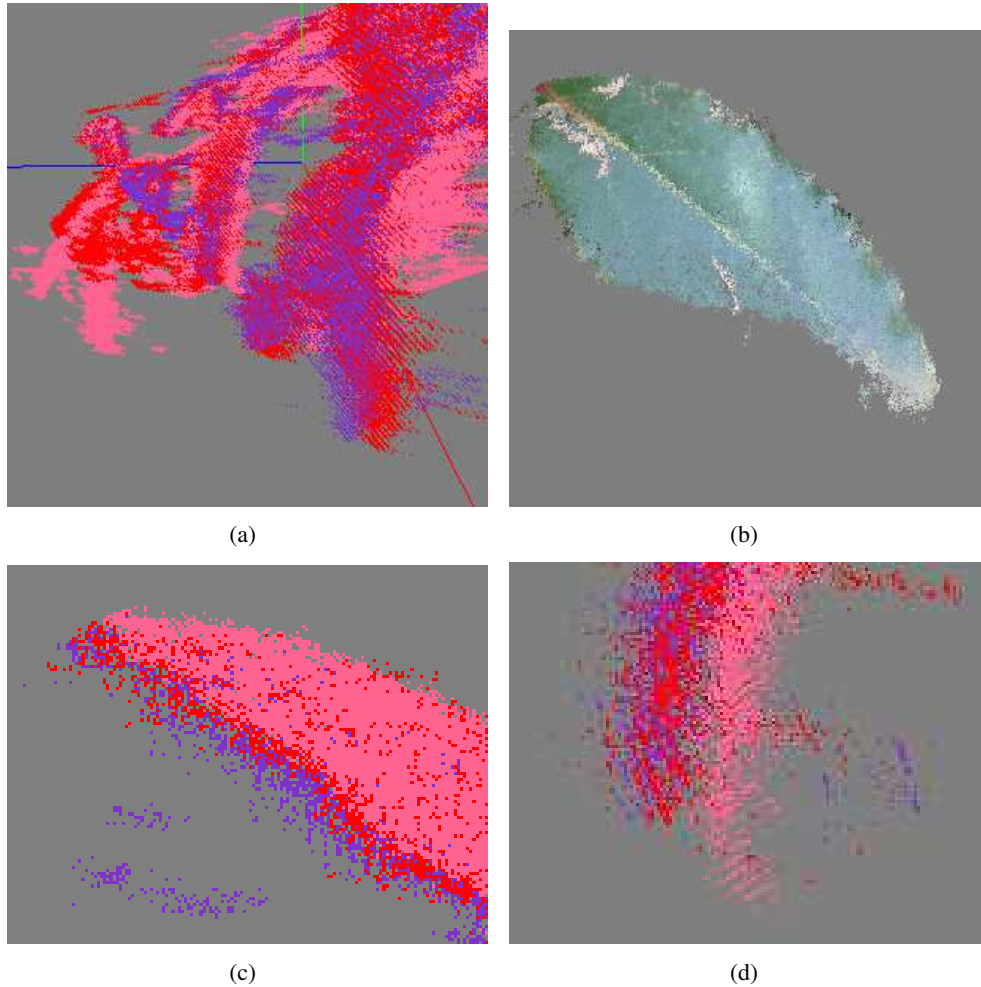


FIGURE 1.6.5 – (a) Problème de superpositions sur 3 nuages, en rouge, rose et violet. (b) Feuille test. En (c) nous pouvons apercevoir que la superposition des nuages à la base de la feuille n'est pas mauvaise contrairement en (d) où au bout de la feuille il n'y a pratiquement aucune superposition. Nous pouvons en déduire qu'un des problèmes sur ce jeu d'images est lié aux mouvements des feuilles et donc qu'il est normal que les nuages ne se superposent pas.

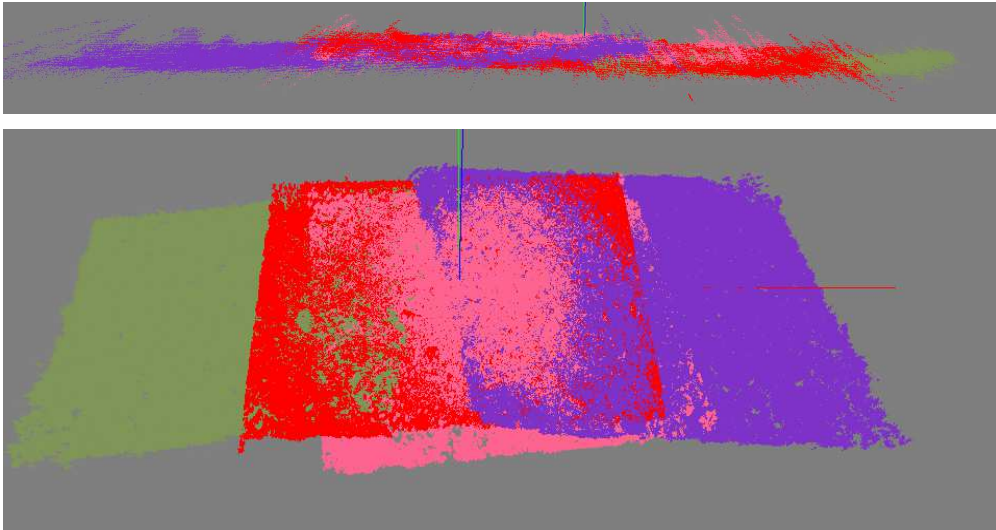


FIGURE 1.6.6 – On peut voir sur ces figures que la superposition dans le cas de nuages de galets n'est pas "mauvaise", mais qu'il reste tout de même des imprécisions et des outliers.

Chapitre 2

Traitement 3D

Introduction

L'utilisation de la chaîne de reconstruction 3D présentée dans le chapitre précédent nous impose de faire une étape de traitement des nuages de points 3D.

Ainsi, nous aborderons les problèmes liés au filtrage de nuages de points 3D. Les nuages n'étant pas exploitables en l'état, ces étapes de traitement sur les nuages seront nécessaires pour l'exploitation de données mesurables. Les nuages produits à l'étape précédente présentent trois problèmes auxquels nous tenterons d'apporter des solutions :

- La présence d'outliers.
- La redondance de points.
- La présence d'un bruit sur les surfaces.

Quelques algorithmes existent pour répondre à certain de ces problèmes. Malheureusement, différentes contraintes ne vont pas nous permettre d'utiliser ces algorithmes. La principale limite est le nombre de points. La plupart de ces algorithmes utilisent comme données des points obtenus par levés laser, que ce soit par un scanner 3D ou par un LIDAR. Or dans ces nuages, le nombre de points n'excède que rarement le million de points 3D, même si récemment des algorithmes gèrent des nuages de points 3D de très grande taille. Nos images, de taille variable de 3 à 20 millions de pixels produisent par couple, avec un recouvrement d'au moins 60 pourcent, un nuage de points de taille allant de 1,5 à 12 millions de points. La taille des nuages rend leurs manipulations compliquées, même si une parallélisation des algorithmes accélèrent de manière spectaculaire l'exécution des algorithmes.

Le nombre de point ne peut pas être inférieur pour une bonne précision des mesures. Ce problème de taille limitera l'utilisation d'algorithmes de filtrage classiques, pour des raisons de taille mémoire et de temps.

Pour supprimer les outliers, l'état de l'art nous fournit des pistes. Mais en étudiant ceux-ci nous nous sommes aperçus que ces algorithmes n'étaient exploitable que pour un petit nombre d'outliers, ce qui n'est pas notre cas. Ainsi, le premier algorithme que nous présenterons permettra la suppression d'outliers par analyse en composantes connexes. Cet algorithme fut créé à l'origine pour le clustering de points 3D. Il identifie le nombre de voisins dans un rayon donné. Nous ajouterons à cette phase,

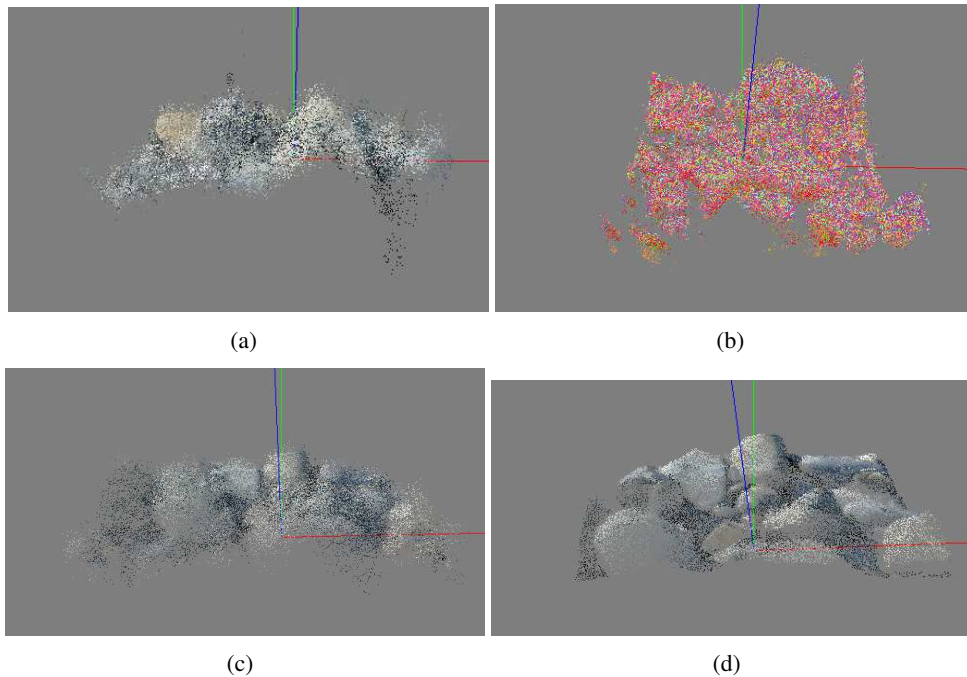
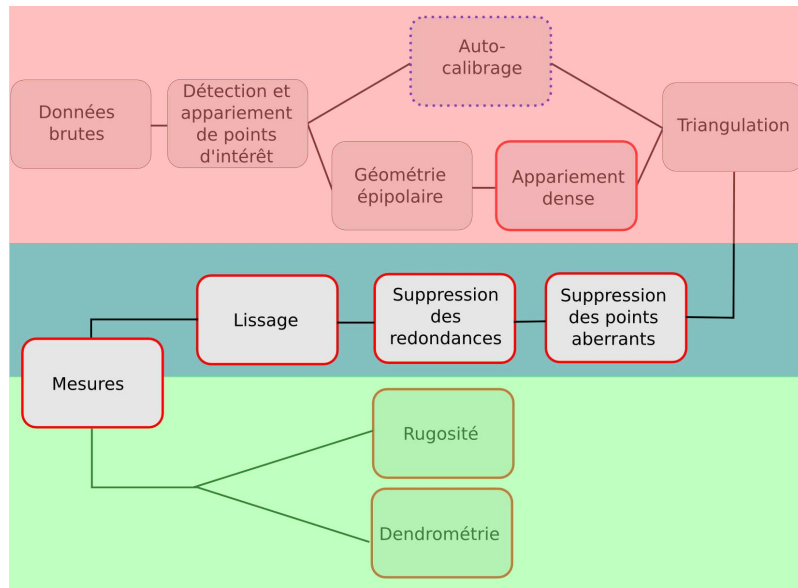


FIGURE 1 – Schéma des problèmes de la chaîne de reconstruction 3D, (a) des points aberrants, (b) des points redondants et (c) d'un bruit de surface, pour obtenir, une fois supprimés, le nuage final vu en (d).

une étape qui supprime les points n'ayant pas assez de voisins.

Le deuxième algorithme que nous présenterons visera à supprimer tous les points redondants dans les nuages de points. Or, nous n'avons pas pu trouver d'état de l'art permettant d'effectuer cette étape. Or, cette étape est nécessaire car elle laissera le nuage moins volumineux et plus précis. Cette suppression se fera donc par un algorithme de corrélation colorimétrique sur les images. Le meilleur point parmi tous sera ensuite choisi. Des choix très forts seront faits pendant cette étape, en particulier sur la définition de la notion de redondance, à partir de laquelle certains points seront considérés comme redondants même s'ils ne le sont pas.

Cependant un bruit persistera et un troisième algorithme sera utilisé pour lisser le nuage. Dans ce domaine, de nombreux algorithmes existent. Le plus performant projette les points sur des surfaces estimées par moindres carrés glissants ou "moving least square", offrant aux nuages un bien meilleur rendu visuel et permettant des mesures plus fiables.



2.1 Suppression des outliers par analyse en composante connexe

Filtrer des nuages 3D n'est pas simple. Ces nuages ne comportent pas seulement du bruit mais aussi de nombreux outliers, c'est-à-dire des points isolés et loin de leur position correcte. Généralement, ces outliers viennent d'une erreur de mise en correspondance. Ces points sont souvent éloignés spécialement et possèdent peu de voisins. Nous tenterons donc de nous servir de cette particularité pour filtrer ces points.

Dans ce cadre nous avons choisi un algorithme d'analyse en composantes connexes. Cette méthode calcule, en chaque point 3D, le nombre de points voisins qui sont contenus dans une boule d'un certain rayon. Seuls les points ayant un nombre de voisins suffisants seront considérés comme inliers, les autres seront supprimés.

Le calcul du voisinage n'est pas simple. L'idée première serait tout simplement de parcourir pour chaque point l'intégralité du nuage et de calculer la distance à chaque point. Or cette méthode exhaustive est bien trop lente, la complexité en $O(n^2)$ ne supporte pas un nombre de points trop grand, comme c'est le cas ici. Ainsi nous avons choisi d'utiliser des algorithmes basés sur le parcours d'arbres de recherche, de complexité en $O(n \log(n))$.

2.1.1 Calcul des k-plus proches voisins

Ainsi, la première étape de cette méthode consiste à identifier en chaque point quels sont ses voisins. Nous avons choisi d'utiliser des algorithmes servant notamment pour le clustering comme [Arya et al., 1998]. Ces méthodes offrent de très bons résultats en terme de mémoire et de rapidité.

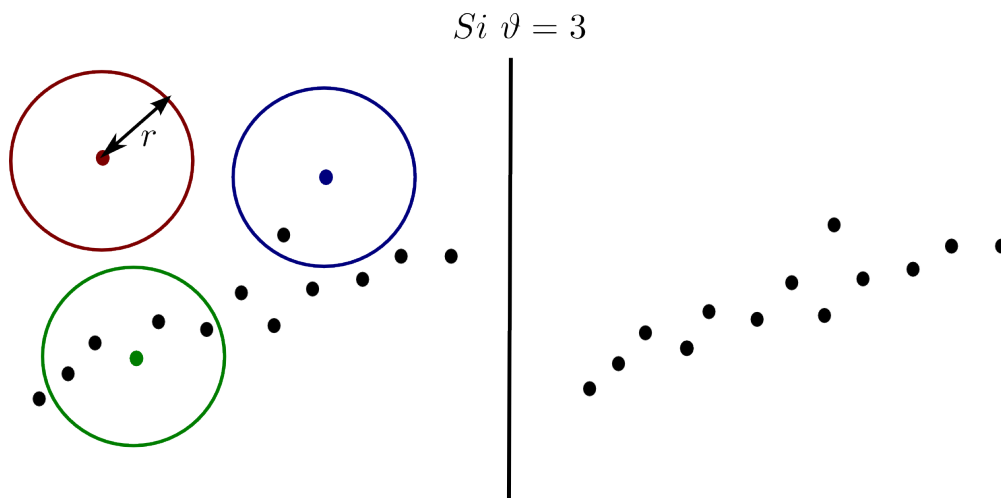


FIGURE 2.1.1 – Schéma de l’algorithme d’analyse en composantes connexes. Ici pour un voisinage de taille r , un nombre ϑ de voisins on s’aperçoit que le point rouge est considéré comme abérrant.

Pour notre approche sur de nombreux nuages de points nous avons préféré des méthodes basées sur les kd -tree [Friedman et al., 1977], approchant très rapidement les k -plus proches voisins [Maneewongvatana and Mount, 2002]. L’algorithme place les points 3D dans un kd -tree et cherche les k -plus proches voisins en utilisant une distance euclidienne.

Enfin une fois les k -plus proches voisins calculés et triés, une recherche est effectuée pour trouver le nombre de voisins qui sont à une distance inférieure à un seuil r , ce nombre peut être équivalent à k . Les valeurs k et r permettront de définir la sensibilité de l’algorithme. Plus r est petit et plus le nombre de voisins sera petit, les points seront donc plus filtrés. En augmentant r , il y aura moins de filtrage. La valeur de k sera bien évidemment choisie en fonction de ϑ le nombre minimum de voisins nécessaire.

Pour des raisons de simplicité, il est préférable d’utiliser $\vartheta = k$.

kd-tree

Les kd -trees sont des structures de données basées sur une subdivision récursive de l’espace en régions hyperrectangulaires appelées cellules. Chaque nœud de l’arbre est associé à une région B , appelée boîte, et est associé à un jeu de points appartenant à cette boîte, comme la zone en gris en 2.1.2. Le nœud racine de l’arbre est quant à lui considéré comme une boîte englobante comprenant tous les points.

En considérant un nœud de l’arbre aléatoirement ; tant que le nombre de points dans la boîte est plus grand qu’une quantité considérée petite, appelé taille du tampon, la boîte est divisée en deux boîtes par un hyperplan orthogonal aux axes de la boîte originale. Trouver cet hyperplan est difficile et il existe de nombreuses règles de divisions qui déterminent comment l’hyperplan est choisi. On rappellera qu’en 2D l’hyperplan séparateur est une droite, en 3D un plan.

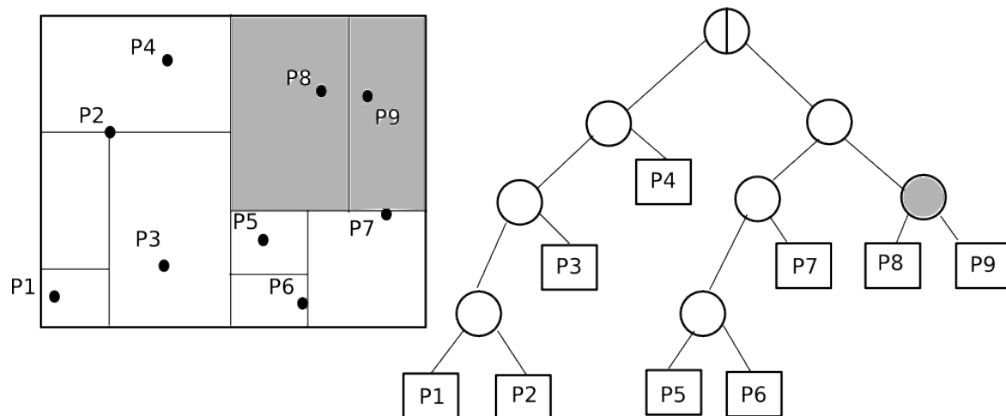


FIGURE 2.1.2 – Schéma du modèle de kd-tree 2D avec une taille de tampon de taille 1. Avec en gris la correspondance entre l’arbre et les zones de divisions sur le nuage.

Une fois la division en deux de la boîte effectuée, les points appartenant à l’hyperplan sont associés à l’une ou l’autre des deux boîtes filles créées. Si la boîte ne peut plus se subdiviser, alors la boîte devient une feuille de l’arbre. Une boîte ne peut plus se subdiviser si elle contient un nombre inférieur ou égal à la taille de tampon. A la fin chaque feuille contient un nombre de points inférieur ou égal à la taille de tampon.

Ainsi l’arbre se construit à travers la séparation récursive des boîtes en deux boîtes filles. Chaque boîte parent sera considérée comme un nœud et chaque boîte indivisible comme les feuilles de l’arbre.

Le problème reste de savoir comment séparer les boîtes en deux boîtes filles. Pour cela de nombreuses règles existent, pour un sous-jeu de données à séparer S de n points 3D :

- la règle standard de séparation, dans laquelle la dimension de séparation est celle correspondant à l’écart maximum de S . Le point de séparation est considéré comme la médiane des coordonnées dans cette dimension. Une partition de l’espace médian est alors effectuée. Cette partition garantit que la taille de l’arbre est de l’ordre de $\log(n)$ et de taille $O(n)$. Mais malheureusement ne garantit pas une uniformité de la taille des cellules.
- la règle du mid-point, dans laquelle la coupure se fait dans chaque cellule au milieu du plus long côté selon l’orthogonale de celui-ci. Malheureusement celui-ci peut entraîner des coupures triviales où presque tous les points appartiendront à un seul côté de la séparation, provoquant un déséquilibre de l’arbre augmentant ainsi la complexité de recherche. Une variante permet d’éviter cet écueil en utilisant un mid-point flottant, permettant à l’hyperplan de coupure de pouvoir se déplacer. Cette méthode sera choisie pour sa simplicité, son coût peu élevé et le fait qu’elle permet d’éviter les coupures triviales. La taille maximum de l’arbre est finalement en $O(n)$ et la profondeur maximum de l’arbre est n .
- la règle de la coupure juste, dans laquelle on cherche un compromis entre les deux méthodes citées ci-dessus montre aussi de très bons résultats. Mal-

heureusement dans certains cas dégénérés cette méthode peut entraîner des coupures triviales. Une variante glissante existe de même. Mais la simplicité de la méthode précédente nous l'a fait choisir, pour des résultats in-fine sensiblement équivalents.

Une fois la règle de coupure et la taille de tampon (choisie à 1 dans notre chaîne) connues, l'arbre est construit. Mais il reste à connaître la méthode de recherche des voisins.

méthode de recherches des voisins dans le kd-Tree

L'algorithme de recherche de voisins d'un point se fera par la méthode standard de recherche [Friedman et al., 1977], qui visite chaque nœud dans un ordre hiérarchique, en commençant par le nœud racine de l'arbre de recherche. L'algorithme utilisé est une adaptation pour la recherche approchée des voisins. l'algorithme descend l'arbre récursivement, quand l'algorithme trouve la première feuille, elle est considérée comme point voisin temporaire. L'algorithme effectue alors de manière récursive pour tous les nœuds les opérations suivantes :

- si le nœud est une feuille, on teste la distance du point contenue dans la boîte au point requête. On cherche à insérer cette distance dans le tableau des distances, s'il trouve une place, on décale le reste du tableau, s'il ne trouve pas sa place, c'est que ce n'est pas un voisin.
- sinon :
 - le nœud enfant le plus proche du point requête est visité. La distance d'une boîte à un point est calculée exactement par une distance incrémentale mise-à-jour comme décrit dans [Arya and Mount, 1993].
 - L'autre enfant est visité si la distance du point requête à la ligne de séparation de la boîte mère est inférieure à la distance du point requête aux voisins, c'est-à-dire que l'hyperplan intersecte l'hypersphère de rayon égal à la plus grande distance. Sinon ce côté est supprimé de la recherche.

Ainsi on descend l'arbre de recherche tant que le nombre de voisins k n'est pas atteint. L'intérêt d'une telle méthode est la rapidité de recherche des voisins dans l'arbre la complexité étant en $O(\log(n))$. L'algorithme peut être synthétisé comme expliqué dans l'algorithme 2

2.1.2 Analyse en composantes connexes

Une fois l'arbre de recherche connu et les k -plus proches voisins trouvés et triés, il suffit à partir d'un rayon de taille r de compter le nombre de voisins dont la distance euclidienne au point est inférieure à ce radius et ainsi de vérifier que le point à un nombre de voisins supérieurs au nombre minimum de voisins ϑ .

Notre algorithme utilise ainsi une règle simple :

$$V'(p) = \{p' \in V(p) \mid d(p', p) < r\} \quad (2.1)$$

$$ACC(\Omega) = \{p \in \Omega \mid card(V'(p)) > \vartheta\} \quad (2.2)$$

Avec Ω l'ensemble des points 3D, p un point 3D requête, $V(p)$ le voisinage de p comprenant les k -plus proches voisins, $V'(p)$ le voisinage de p situé à une distance

Algorithm 2 Recherche(Point PR, Noeud n) :

```
Initialisation des tableaux des distances, avec pour distance maximale
FLOAT_MAX, et des voisins
if Le nœud est une feuille then
    if la distance du point requête au nœud  $n$  est inférieur à la plus grande distance du
    tableau then
        On trouve sa place dans le tableau des distances ;
        Mise à jour des tableaux de distances et de voisins ;
    end if
else
    Le nœud correspond à une boîte ;
    if l'hyperplan ne coupe pas l'hypersphère then
        if le point se trouve à gauche ou en haut de l'axe, on cherche alors dans le nœud
        fils de gauche then
            retour récursif de "Recherche" à gauche du nœud ;
        else
            retour récursif de "Recherche" à droite du nœud ;
        end if
    else
        l'hyperplan coupant l'hypersphère, il faut parcourir les deux côtés du nœud
        récursivement ;
    end if
end if
return Tab_vois ;
```

inférieur à r , d la distance euclidienne, r le rayon maximum.

La simplification consistant à $\vartheta = k$ permet de vérifier uniquement si la distance, de p au k ème voisin, est inférieure à r .

Les résultats montrés dans les figure 2.1.3 à 2.1.6 sont des exemples sur un seul des nuages obtenus, mais la méthode est appliquée de manière automatique à tous les nuages.

On remarque sur les résultats présentés qu'un grand nombre de points sont supprimés. Ce phénomène est lié à la présence de beaucoup d'outliers et d'un choix de paramètres délibérément stricts, quitte à supprimer certains points bien placés mais isolés. Nous pouvons faire ce choix grâce à la présence de nombreuses redondances d'informations.

La présence de seconds plans parfois éloignés peut aussi entraîner un filtrage important, surtout quand la prise de vue se fait de manière rasante, c'est-à-dire très peu fronto-parallèle. On peut observer ce phénomène dans les images de végétation, où la présence du second plan est gênant car trop éloigné par rapport à la ligne de base. La triangulation est alors peu précise et on obtient alors ce phénomène de "queue de comète".

Les points ainsi reconstruits seront éloignés les uns des autres malgré des différences de disparité inférieures au pixel. Leurs dispersions dans l'espace entraînent une suppression de ces points considérés par l'algorithme comme des outliers même s'ils

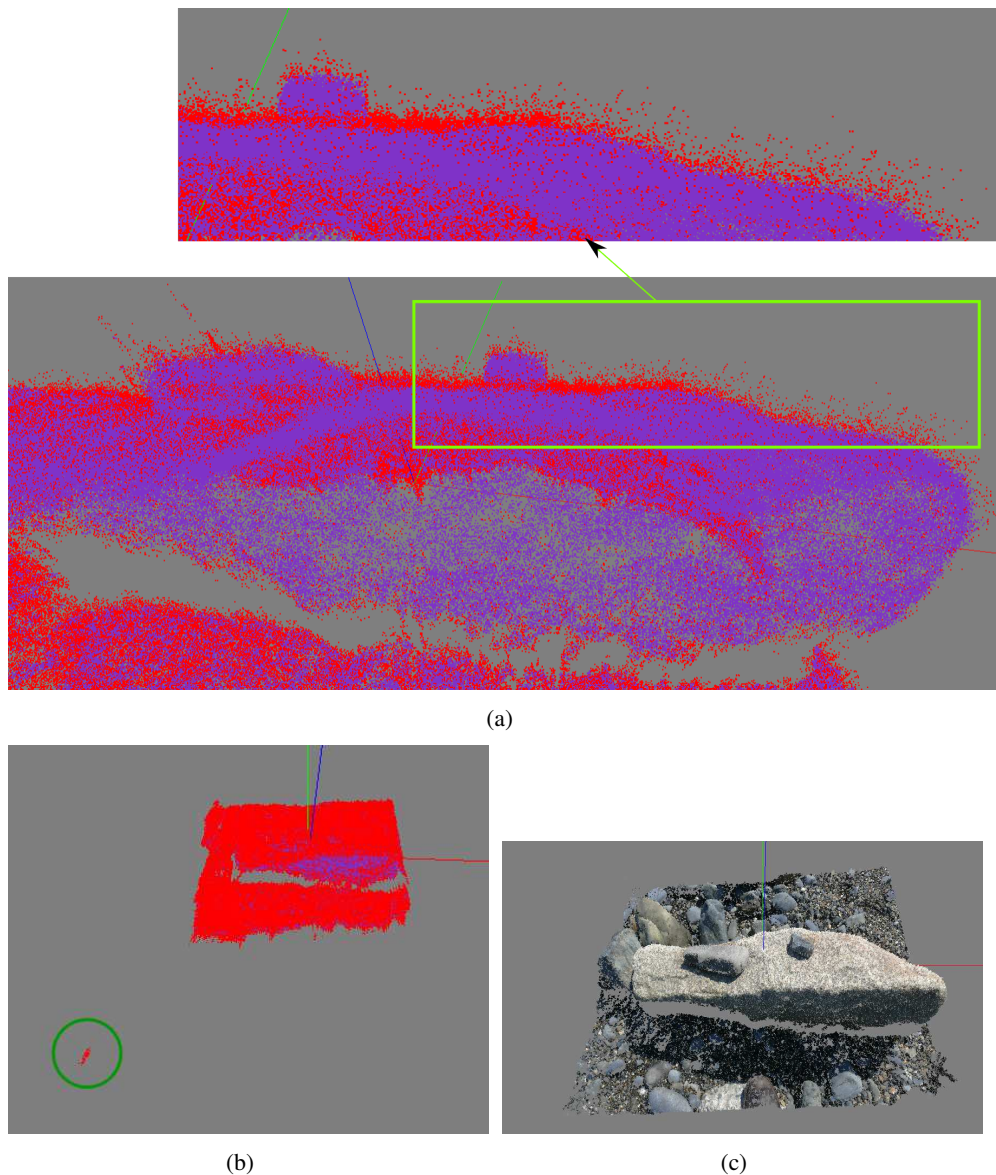
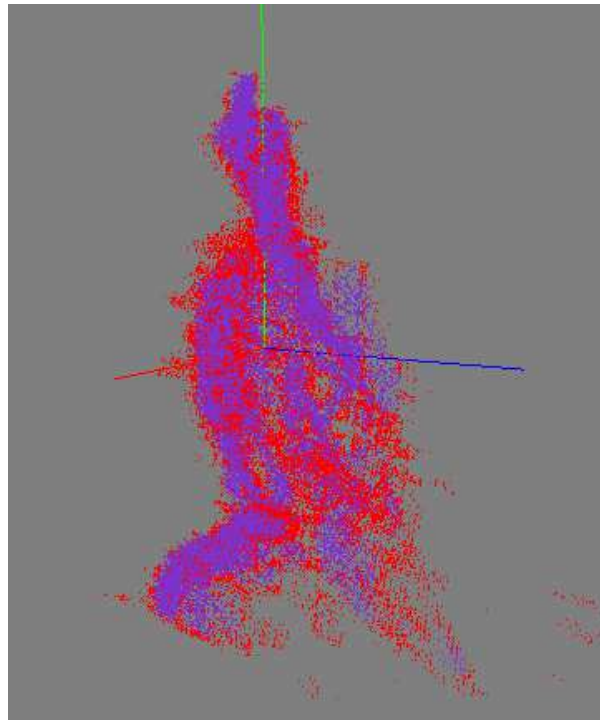
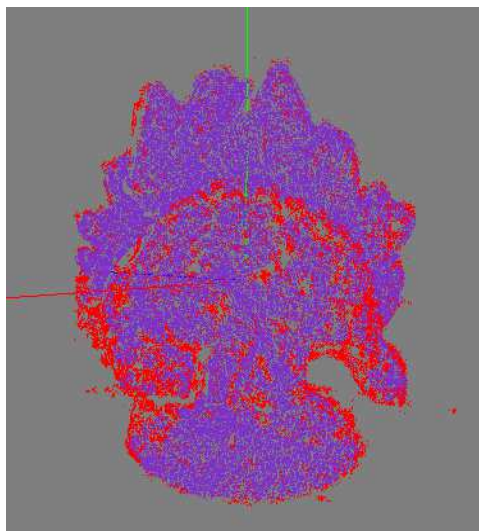


FIGURE 2.1.3 – (a) Coupe du nuage de points sur les galets et le caillou, avec en rouge : les points du nuage original qui seront supprimés, et en violet : les points du nuage original qui seront gardés dans le nuage filtré. (b) Dans le cercle vert, on peut voir qu'un petit nuage de points, correspondant à des points mals appariés, est bien considéré comme outliers malgré leurs cohérences locales. (c) Vue du nuage filtré en couleur original, on peut voir qu'on conserve assez de points pour maintenir toutes les informations de distances.

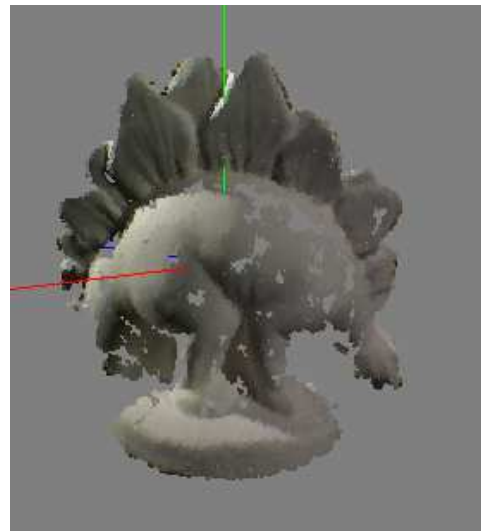
ne devraient pas l'être. Le manque de précision sur les points éloignés implique leur filtrage à cette étape mais ce n'est finalement pas un problème car leur trop grande imprécision quand à leur position réelle dans l'espace implique qu'ils ne seraient de toutes les façons pas exploitables pour les mesures. Ainsi on peut considérer que dans



(a)



(b)



(c)

FIGURE 2.1.4 – (a)(b) En rouge les points supprimés du nuage original et en violet les points gardés dans le nuage filtré obtenus à partir des images de dinosaure fournit par [Scharstein and Szeliski, 2002]. (c) Vue du nuage filtré avec l'information de couleurs venant des images.

notre problématique qui est le calcul de mesures, sur des nuages de points 3D, il est important de les supprimer.

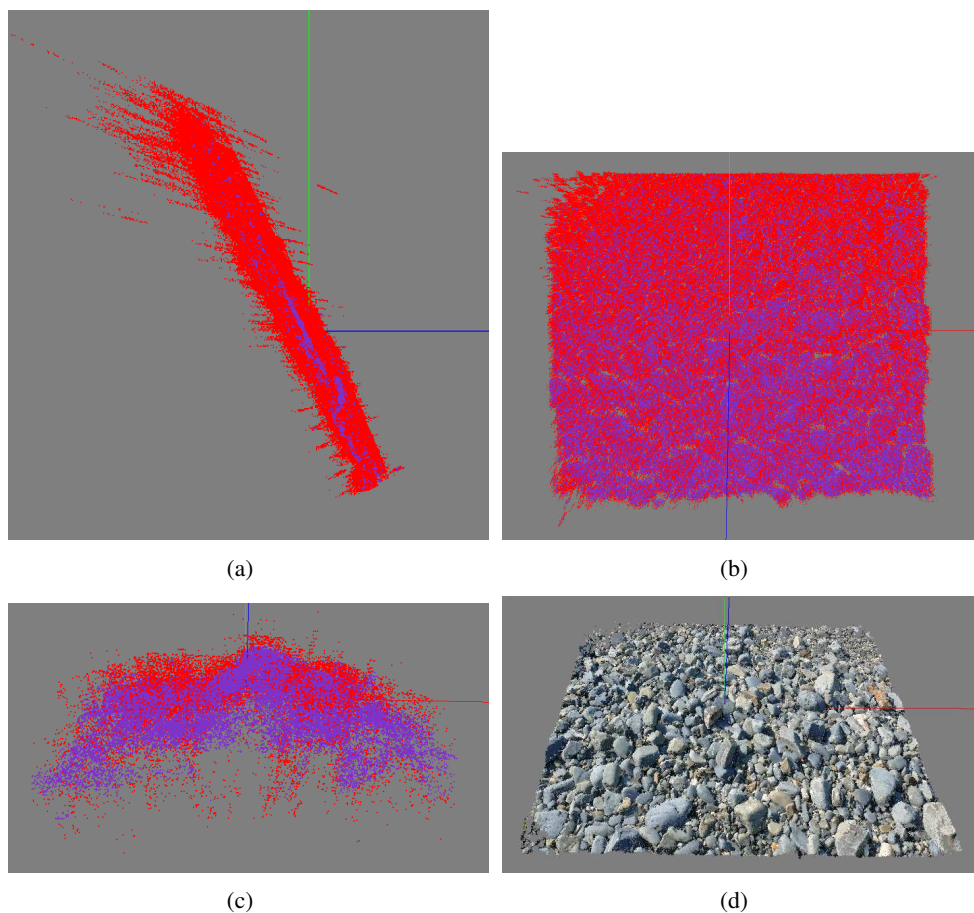


FIGURE 2.1.5 – (a)(b)(c) Vues des nuages originaux et filtrés avec en rouge les points supprimés sur le nuage original et en violet les points gardés, on remarque ainsi que la prise de vue n'étant pas fronto-parallèle mais rasante les points les plus éloignés possèdent le plus grand nombre d'outliers (phénomène particulièrement évident en (b) où les points rouges se concentrent majoritairement sur le haut de l'image, soit les points les plus éloignés des caméras). (d) Vue du nuage filtré en couleur réelle.

Dans le premier cas traité, c'est-à-dire le nuage de points de galets vu sur la figure 2.1.3, on remarque dans le grossissement que la plupart des outliers sont supprimés à cette étape. Ainsi, le pourcentage moyen de points supprimés est de 33,2%, ce qui veut dire que pour des nuages de 1.5 millions de points en moyenne, le nombre de points gardés est en moyenne d'un million. Ainsi, il reste tout de même 18 millions de points en tout, car il a été pris 19 photos, la reconstruction 3D se faisant par couple successif. Dans le tableau 2.1 un récapitulatif des exemples avec le taux de suppression d'outliers et la taille finale moyenne de chaque nuage.

Une fois cette étape de filtrage effectuée, les nuages sont mis en commun. Pour supprimer les points doublons, une étape filtrage colorimétrique sera effectuée. Cela

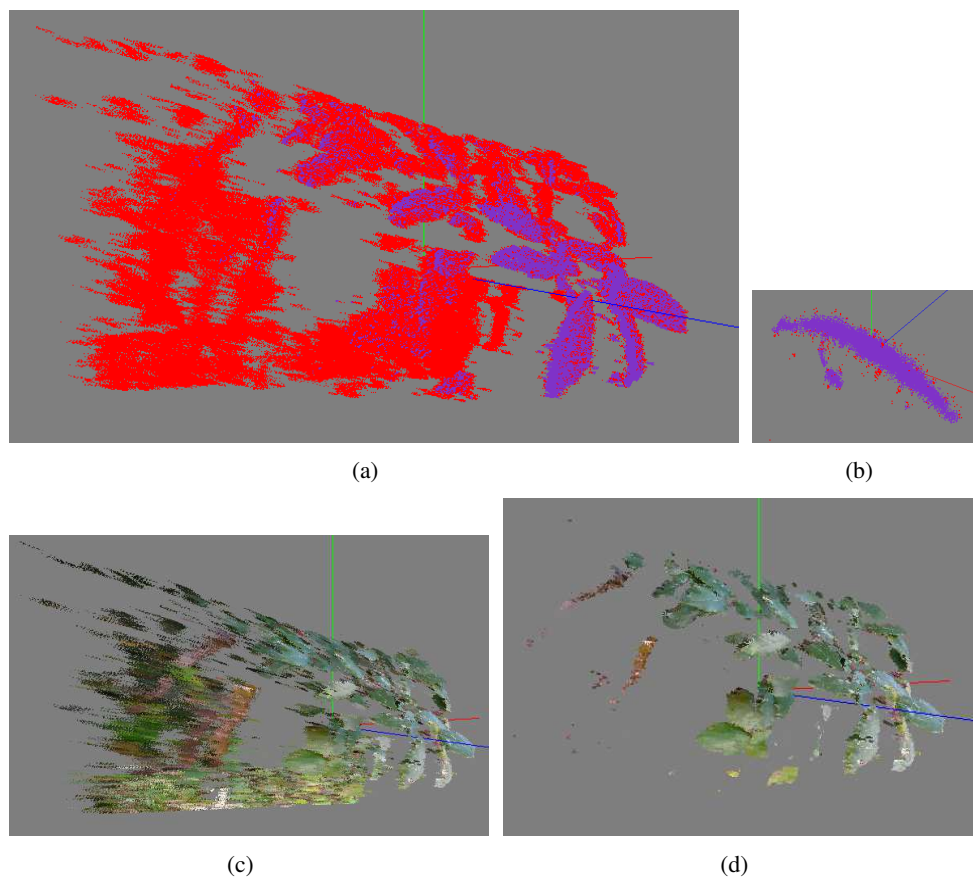


FIGURE 2.1.6 – (a) En rouge le nuage original et en violet les points correspondant au nuage filtré, on remarque que les points gardés sont surtout sur le premier plan. (b) Exemple sur une feuille en particulier. (c) Nuage original, comparé au nuage filtré en (d) confirmant la supposition précédente : une disparité trop faible (éloignement trop grand) par rapport à la ligne de base entraînent que ces points ne sont plus assez précis pour être considérés comme de bons points.

Exemple	nombre de point originel	% supprimé	taille finale du nuage
Caillou	1.5M	33.2	1M
Dinosaure	90k	33.3	70k
Galets	1.8M	20	1.4M
Feuilles	1.2M	10-50	720k

TABLE 2.1 – Résultats du filtrage. Pour les images de feuilles, les résultats ont été séparés entres images prises avec et sans beaucoup d’arrière plan. Provoquant, dans le premier cas, un taux de suppression approximativement de 50% alors que pour le deuxième cas, seulement 10% des points sont filtrés. La moyenne sur l’ensemble des images est de 33% de points supprimés.

permet de vérifier la non-redondance de l’information et rendre ainsi le nuage cohérent.

2.2 Suppression des points redondants

Nous avons à cette étape de la chaîne de traitement, un certain nombre de nuages filtrés. Malheureusement, une redondance très forte de points existe rendant la manipulation des nuages difficile. La redondance vient du fait que pour un pixel d'une image, on peut obtenir plusieurs points 3D, qui eux n'ont pas les mêmes coordonnées 3D et sont donc considérés comme des points différents. Ces points peuvent parfois être très éloignés les uns des autres. Réunir tous les nuages de points filtrés ensemble est alors impossible, étant trop volumineux en taille. En plus de la redondance, un problème de superposition rend les surfaces extrêmement épaisses et rend toute mesure difficile. Ainsi, il faut effectuer une étape permettant de supprimer ces redondances et épaisseurs, sans perdre la cohérence.

2.2.1 Recalage des nuages

Différentes techniques sont envisageables comme effectuer une étape de recalage entre les nuages pour pouvoir les superposer. Un algorithme recalant efficacement les nuages permettrait de supprimer la redondance (pour un point recalé sur l'autre, nous n'en garderons qu'un seul). Ainsi des techniques comme un recalage par ICP [Besl and McKay, 1992] qui reste standard pour faire correspondre deux nuages de points peuvent être utilisées.

Mais les tests effectués ont été erronés, lié à la sensibilité au bruit de la plupart des techniques de recalage. Or, nos nuages possèdent beaucoup de bruit. Ce bruit provenant notamment des différentes estimations faites, à la corrélation sub-pixélique et à la triangulation. De plus le volume de données ne permet pas une utilisation aisée et rapide de ces algorithmes.

L'autre explication permettant de comprendre les mauvais résultats vient du fait que ces techniques ont été développées pour des nuages obtenus par l'utilisation d'un laser ou du LIDAR et particulièrement pour la reconstruction d'objets d'art. Le scanner 3D permet la reconstruction des nuages avec peu de bruit, et le recalage devient alors aisé notamment par des variantes de l'ICP [Rusinkiewicz and Levoy, 2001]. Mais dans notre cas, et particulièrement dans le cas des feuilles, les algorithmes testés n'ont donné aucun résultat satisfaisant, voir complètement faux.

Une idée serait de lisser les nuages avant de tenter de les recalcr, mais le bruit persistant nous fait obtenir des lissages très différents en fonction des nuages, empêchant un recalage correct. Finalement, la superposition originale est restée meilleure que celle obtenue par recalage.

Nous avons donc préféré abandonner l'idée d'utiliser une technique de recalage.

2.2.2 Suppression des redondances par filtrage colorimétrique

La deuxième idée qui nous a paru évidente est un filtrage colorimétrique, mais cette technique n'aurait pas supprimé les problèmes de redondance.

L'impossibilité d'utiliser des techniques de la littérature vient en grande partie du fait que notre approche pour la mesure est originale. La stéréo-vision est plutôt utilisée

pour la reconstruction de scènes fixes comme des bâtiments ou des objets et utilisent généralement un bon calibrage.

Dans le cas où un auto-calibrage est effectué, on ne retrouve dans la littérature aucune mention d'utilisation de stéréo-vision. Ainsi, mélanger auto-calibrage et stéréo-vision est quelque chose d'original et il est donc normal d'être confronté à des problèmes dont on ne trouve pas de solution dans l'état de l'art.

La seule possibilité restante pour tenter de supprimer le bruit et la redondance des nuages 3D est une approche originale, en se servant au maximum des informations aussi bien dans l'espace 3D que dans les images. Nous tenterons donc d'expliquer la méthode développée et les raisons qui nous ont poussées à choisir les techniques employées.

L'idée directrice fut la connaissance des nuages de points 3D, l'image dont ils sont issus, correspondant à l'image gauche du couple, et de la position des caméras. Le fait d'avoir créé un nuage de points 3D couple par couple avec un recouvrement sur les images d'au moins 75%, nous permettait pour chaque pixel d'avoir au moins trois points 3D correspondant. L'idée est alors de ne garder qu'un seul point 3D pour chaque pixel.

Pour chaque point 3D se projetant dans le même pixel d'une image, un score de corrélation est calculé. Cette corrélation est calculée en reprojectant ces points dans toutes les images proches de celle à laquelle appartient le pixel et de choisir le meilleur score, celui-ci remportant le tout (winner take all), les autres points étant supprimés. L'algorithme peut se résumer en trois étapes :

- Pour chaque pixel de l'image, il faut trouver tous les points 3D se projetant dans le même pixel comme expliqué dans la figure 2.2.2.
- Pour chaque point 3D, faire un calcul de corrélation en reprojectant ce point dans toutes les images par lesquelles le point est vu comme expliqué à la figure 2.2.3.
- Une fois les corrélations calculées, choisir le meilleur point et supprimer les autres.

Cette technique bien que simple offre un très bon rapport résultat/rapidité. De plus, la corrélation étant subpixelique plusieurs points 3D peuvent se projeter dans les mêmes pixels mais en restant correct. Dans le but de simplifier la méthode, plusieurs heuristiques seront faites. La première est de ne pas calculer tous les scores de corrélation mais de s'arrêter au premier inférieur à un certain seuil. Cette heuristique transforme l'algorithme WTA en simple seuillage.

La deuxième heuristique consiste à ne pas parcourir tous les pixels de chaque image. Chaque point ayant un score de corrélation inférieur au seuil est reprojecté dans les images par lesquelles il est vu ; les pixels correspondant à ces projections sont masqués. C'est-à-dire que dans la figure 2.2.1, si le point rouge est choisi, alors les pixels, dans les images bleues, ne seront jamais traités.

identification des redondances

Une fois l'image choisie, pour chaque pixel, on est censé parcourir, la liste complète des points 3D pour trouver les points se projetant dans ce pixel. Faire cela sur un nuage de plus de 20 millions est beaucoup trop long, car pour m pixels et n points on obtient une complexité de $O(n.m)$. Il faut donc trouver des heuristiques de

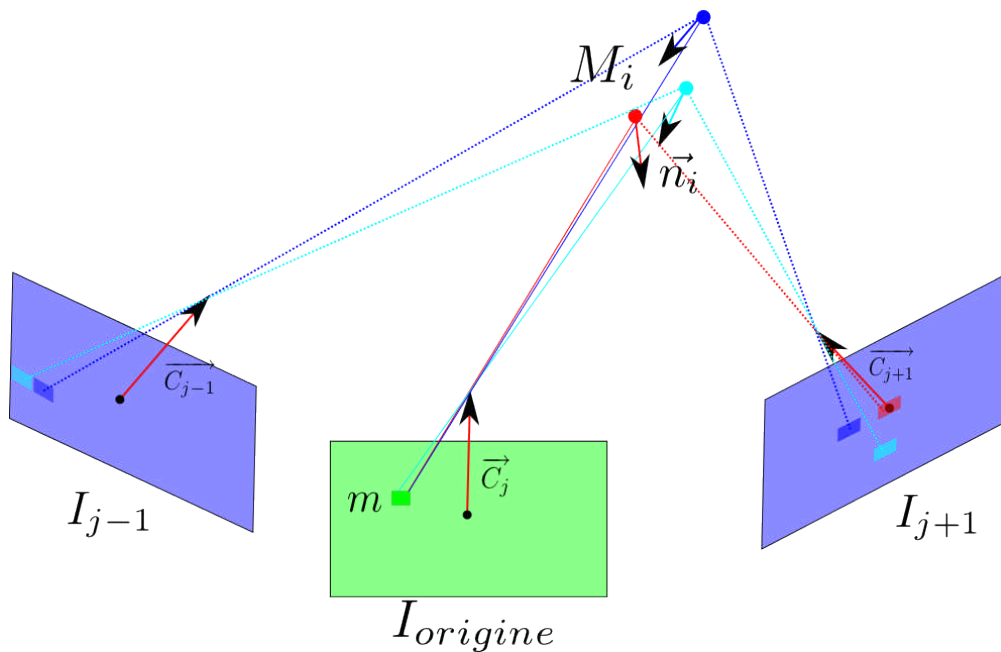


FIGURE 2.2.1 – Schéma général du modèle de suppression de redondance pour un pixel m , des points M_i se projetant dans le pixel m de l'image $I_{origine}$ sont reprojétés dans les images appartenant au cône de projection, ici seuls I_{j-1} et I_{j+1} seront prises en compte pour les points cyan et bleu. Seule I_{j+1} est utilisée pour le point rouge. Une corrélation sera effectuée entre les différents pixels colorés sur les images et le pixel m .

calcul.

L'idée est de conserver un tableau de référence, associant à chaque pixel la liste des points 3D se projetant dans celui-ci. Pour cela il suffit de parcourir le nuage complètement une seule fois et de sauvegarder pour chaque point dans quel pixel il se projette. Ainsi nous ne parcourons le nuage qu'une seule fois par image rendant la complexité en $O(n)$.

Malheureusement cet algorithme ne peut pas être utilisé de cette manière, car certains points sont cachés et ne sont pas vus par cette image. Par exemple si des photos sont prises d'un objet en tournant autour, les points qui appartiennent à la face cachée de l'objet pour l'image ne doivent pas être pris en compte et doivent être ignorés. Pour savoir comment ignorer ces points, il faut vérifier l'angle que fait leur normale, qui sera estimée par le vecteur du point à leur caméra originale, avec l'axe de prise de vue de l'image. Si cet angle est petit, alors le point est pris en compte, sinon il est ignoré. De plus, certains points ne seront pas pris en compte dans cette correspondance, car ils ne se projettent tout simplement pas dans les limites de l'image, dans ce cas là le point est ignoré.

On peut résumer cet algorithme par l'équation suivante :

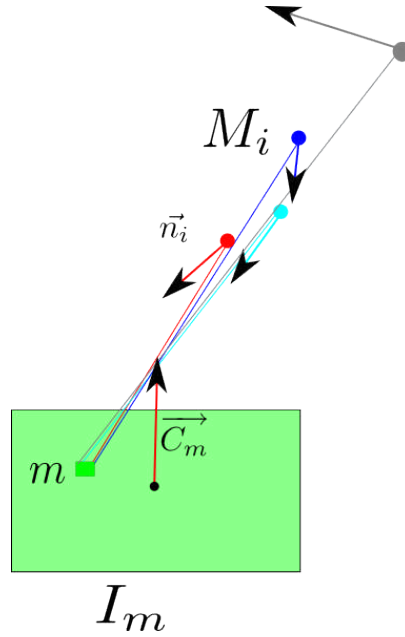


FIGURE 2.2.2 – Schéma du choix des points 3D M_i pour un pixel m de l’image I_m . Sur la figure 4 points se projettent sur m mais seuls 3 appartiennent à ω , car leurs \vec{n}_i sont inférieurs à α : le rouge, le cyan et le bleu. Le point gris a un \vec{n}_i formant un angle trop grand, il n’est pas pris en compte dans ω .

$$\omega = \{ M_i \mid (\widehat{\vec{C}_m, \vec{n}_i}) < \gamma \}$$

Avec ω l’ensemble des points 3D se projetant en m de l’image I_m , \vec{C}_m la direction de prise de vue de I_m tel que $\vec{C}_m = R_m^T [0 \ 0 \ -1]$, avec R_m la matrice de rotation donnée par le calibrage de l’image I_m , M_i les points dont on cherche à savoir s’ils sont vus par l’image traitée et dont le vecteur point à l’image originale gauche est \vec{n}_i et γ l’angle maximum.

Calcul des corrélations et choix du meilleur point

Une fois que l’on connaît l’ensemble des points se projetant réellement dans un même pixel m d’une image, une corrélation est effectuée pour chacun des points $M \in \omega$. On projettera chaque point M dans toutes les images par lesquelles il est vu et un calcul de corrélation aura lieu entre ces pixels et le pixel m comme expliqué dans la figure 2.2.1.

Le calcul de corrélation pour un point M se projetant dans les pixels p des images I_j se fera tel que :

$$corr(M) = corr(p \in I_j \mid (\vec{n}, \vec{C}_j) < \alpha)$$

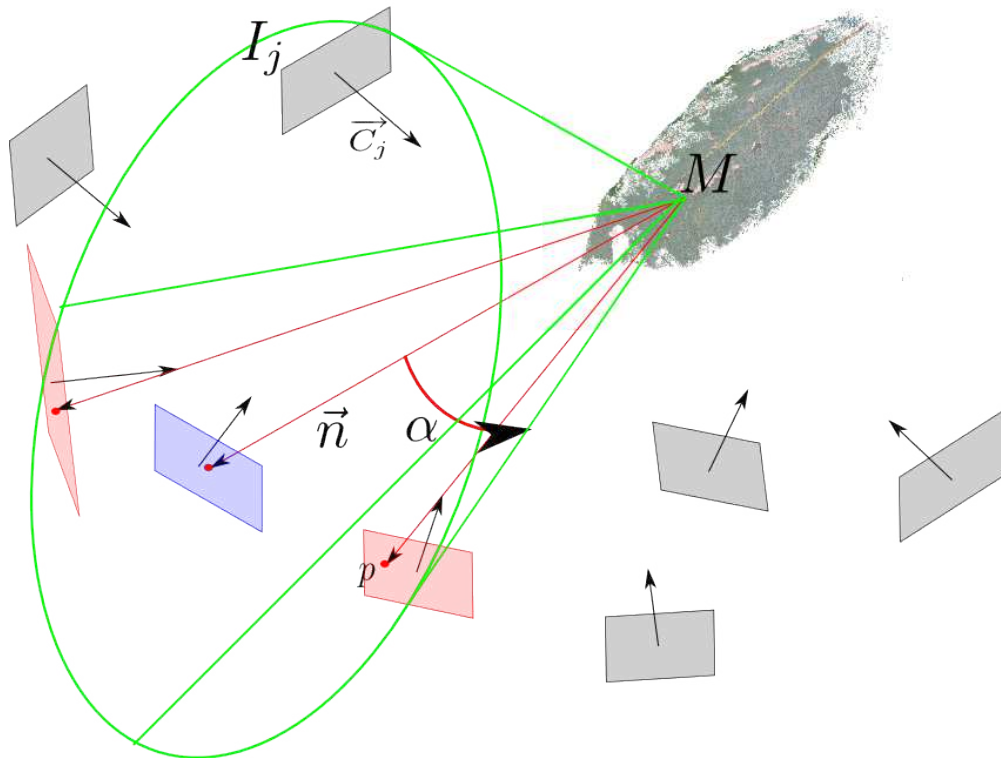


FIGURE 2.2.3 – Schéma de projection pour un point 3D. La corrélation ne se fera que sur les images appartenant au cône d'angle α , c'est-à-dire que le point 3D ne sera re-projeté que dans les images rouges pour faire la corrélation. L'image bleue correspond à l'image originale du point et le vecteur permettant le calcul du cône correspond au vecteur du point à sa caméra originale. Les images grises ne seront pas prises en compte, ainsi si ce point est le meilleur alors un masque sera mis sur les pixels des images rouges mais pas sur celles grises.

Avec C_j le vecteur de prise de vue de I_j tel que $C_j = R_j^T [0 \ 0 \ -1]$, n le vecteur du point M à son image original gauche et $corr$ la différence au carré des couleurs normalisées.

On choisira ainsi le point 3D final M_f tel que :

$$M_f = \min_{i \in \omega} (corr(M_i))$$

Une fois cette étape effectuée de nombreux points ont disparus, et tous les nuages ont perdu leur redondance.



FIGURE 2.2.4 – Nuage final des images de galets et cailloux. Avec cette étape de nombreux points ont disparus car le nuage en lui même ne fait que 2 millions de points.

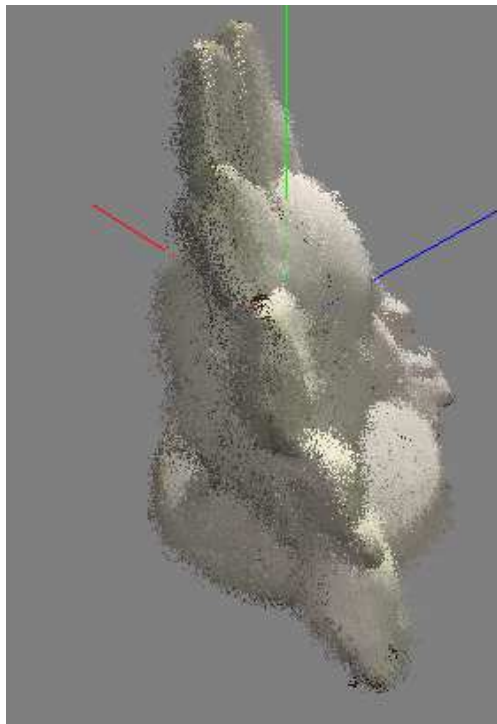


FIGURE 2.2.5 – Nuage final des images de dinosaures.

2.3 Lissage du nuage de points par moindres carrés glissants

Une fois en possession de nuages de points sans redondances d'information, des mesures pourraient être effectuées, mais malheureusement un léger bruit subsiste.

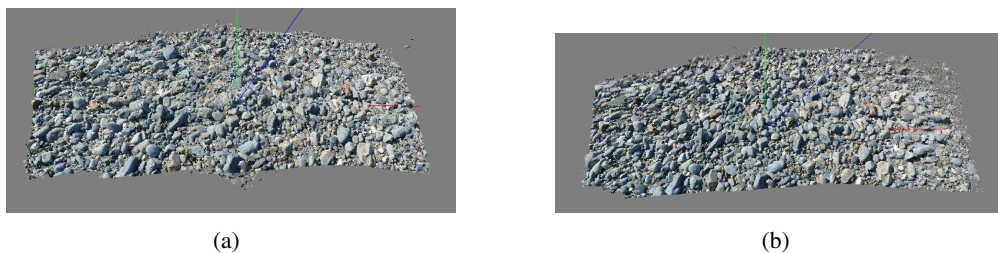


FIGURE 2.2.6 – Résultats comparatifs entre en (a) une vue de 9 nuages sur 34 obtenus de 8 millions de points et (b) 1.5 million de points sur le nuage final une fois les redondances supprimées.

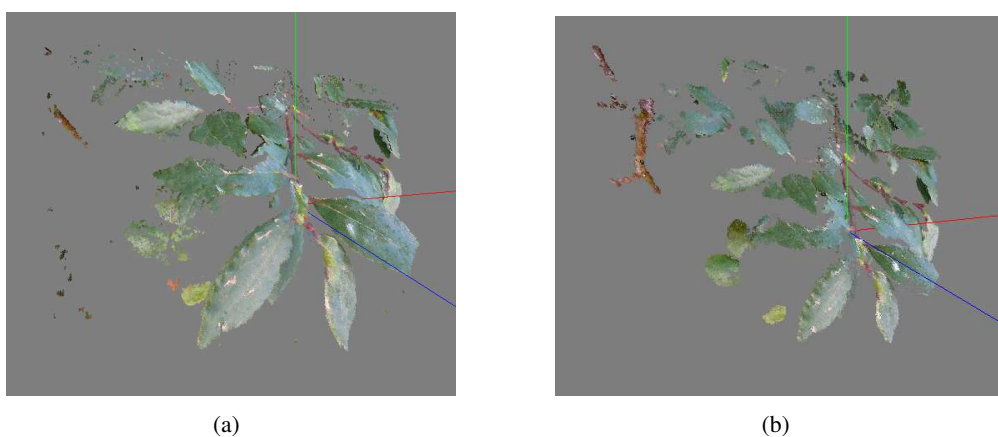


FIGURE 2.2.7 – Résultats comparatifs entre en (a) une vue de 10 nuages sur 22 obtenus de 5 millions de points et (b) 1 million de points sur le nuage final une fois les redondances supprimées.

Ce bruit ne permet pas de faire les mesures avec la précision souhaitée. Ce bruit provoque sur les nuages une impression d'épaisseur sur la surface rendant toute mesure imprécise.

Les techniques pour supprimer le bruit sur un nuage de points 3D furent développées pour aider à la triangulation des nuages de points. Une triangulation tente de transformer un nuage de points 3D en surface maillée et de nombreux algorithmes existent comme la triangulation crust ([Amenta et al., 1998]) ou cocone ([Dey and Goswami, 2003]).

D'autres techniques permettent à la fois de lisser et de trianguler comme la reconstruction de Poisson [Kazhdan et al., 2006] qui cherche les ISO-surfaces en lissant le champ des normales dans un octree et qui a montré de bons résultats.

Mais les techniques qui ont montrés les meilleurs résultats dans le cadre non structuré quant à la suppression du bruit sont les algorithmes basés sur une estimation d'une surface moyenne s'adaptant aux données par moindres carrés pondérés glissants ou

"Moving least square" (MLS).

Dans ce chapitre nous définirons le principe général des surfaces MLS, puis nous parlerons des méthodes de projections, en expliquant l'algorithme que nous avons choisi d'utiliser.

2.3.1 Définition générale des MLS

Ces techniques sont basées sur les principes de Shepard ([Shepard, 1968]) pour le lissage et l'interpolation des données 2D et son adaptation aux surfaces par Lancaster ([Lancaster and Salkauskas, 1981]). L'idée générale est d'utiliser les techniques de moindres carrées pondérées pour des points quelconques et ensuite de les glisser sur tout le domaine des paramètres.

Il est montré que la fonction globale approchant les points peut être obtenue à partir d'un jeu de fonctions locales. Ainsi pour un point x , la fonction globale $f(x)$ est estimée sur un voisinage de points p_i , telle que :

$$f(x) = \underset{f_x \in \Pi_m}{\operatorname{argmin}} \sum_{i=1}^N (f_x(p_i) - f_i)^2 \theta(d(x, p_i)) \quad (2.3)$$

Avec θ une fonction de poids non-négative, Π_m l'ensemble des polynômes de degré m , N la taille du voisinage, f_i le nombre scalaire correspondant à la distance du point à la fonctionnelle en p_i et d la distance Euclidienne. On définit ainsi l'approximation MLS de x tel que $M_f = f_x(x)$ dont une définition équivalente est trouvée par la propriété de reproduction des polynômes est donnée par Levin ([Levin, 1998]). La surface est donc définie par un ensemble de polynômes locaux.

Ces techniques furent réactualisées par leur utilisation dans un nuage de points 3D par Levin ([Levin, 1998]) puis Alexa ([Alexa et al., 2001]). Le but est de trouver une fonction qui décrit la géométrie localement.

Le principe général des MLS défini par Levin est de trouver la fonctionnelle par morceaux qui estime le mieux les données bruitées autour d'un certain voisinage. Il s'agit d'une méthode locale d'estimation. La fonctionnelle utilisée est généralement un polynôme de degré faible (3 ou 4). Pour trouver ce polynôme, il faut tout d'abord trouver l'hyperplan moyen par rapport aux données et en prenant le centroïde des données comme centre du repère.

2.3.2 État de l'art

Le but de cet état de l'art est d'expliquer pourquoi l'utilisation de fonctionnelle avec polynômes, qui était très complexe, a été abandonnée. Cette complexité étant due en grande partie à l'utilisation de minimisation non linéaire, d'autres méthodes itératives de projection, simples, ont été développées pour accélérer le résultat.

Cette méthode a permis l'approximation des nuages de points bruités par un fonctionnelle. Les MLS ou méthodes de moindres carrées pondérés glissants ont été développées plus généralement pour définir les structures manquantes aux points 3D. Cette technique permet aussi de lisser et de débruiter efficacement un bruit de

positions des points 3D.

Alexa avec son article [Alexa et al., 2001] approche les points par une surface MLS et définit cette surface comme un ensemble de points stationnaires d'un opérateur de projection itératif. Il définit que l'estimation de la fonctionnelle est équivalente à l'estimation d'une projection sur un plan de manière itérative. Cet apport est très important car il permet de retrouver simplement et très rapidement la surface MLS.

Ensuite, Amanta en 2004 [Amenta and Kil, 2004] étend cette définition en démontrant qu'il est possible de définir la surface MLS comme une surface implicite des distances pondérées. Il prouve qu'il est équivalent d'estimer le polynôme de degré faible (2, 3 ou 4) dans le plan local de référence dont le centre est x' , que d'utiliser l'approche simple d'Adamson et al en 2003 [Adamson and Alexa, 2003] consistant à projeter itérativement le point sur un plan (polynôme de degré 1). Ainsi, le même résultat est obtenu avec une simple procédure de projection itérative sur un plan qu'avec une approximation de la fonctionnelle.

Enfin en 2004 dans [Adamson and Alexa, 2004] et [Alexa and Adamson, 2004] la surface est définie pour des points possédant des normales accélérant la version précédente. Cet opérateur sera nommé PSS pour "Point Set Surface".

Plus tard il est dit que la surface MLS peut être considérée comme une surface implicite en modifiant légèrement l'opérateur de projection. Il est dit dans [Adamson and Alexa, 2003] qu'à partir de $P = p_i \in V(x)$ l'ensemble des points du voisinage de x , le point bruité à projeter, il est possible de définir la fonction implicite $f : \mathbb{R}^3 \leftarrow \mathbb{R}$, qui estimera au mieux la surface $S = x : f(x) = 0$ et pourra être définie telle que :

$$f(x) = n(x)^T(x - c(x)) = 0 \quad (2.4)$$

Avec $c(x)$ le centroïde local de poids et $n(x)$ la normale approchée calculée comme il sera montré dans les équations 2.9 et 2.6.

Puis en utilisant cette définition, il apparaît qu'utiliser des plans de projection revient à trouver une surface implicite. Pour simplifier, l'idée d'utiliser les surfaces implicites vient du fait qu'il fallait trouver la plus petite distance pour le plan de projection dans le sens de la normale. En utilisant une fonction distance, on s'aperçoit que cela définit une surface implicite. En prenant un champ de normales, la surface implicite à zéro est celle qui approche le mieux les données car c'est celle qui minimise la distance des points à cette surface. La procédure de reconstruction de surface implicite se passe en trois étapes : calcul du champ des normales, approximation de la fonction implicite de distance puis application d'un algorithme de reconstruction de surface comme les marching cubes par exemple [Lorensen and Cline, 1987].

Ainsi, en se servant de la définition expliquée par Fleishman ([Fleishman et al., 2005]), il est apparu que l'estimation de la surface par des plans peut être instable. Cette instabilité est due aux problèmes de définition

des surfaces MLS à des taux d'échantillonnage faibles. Ainsi, il est proposé dans [Guennebaud and Gross, 2007] d'utiliser comme surface de projection des sphères algébriques, car il est montré que le plan est un cas dégénéré des sphères algébriques.

2.3.3 Surface MLS de projections

Comme le montre l'état de l'art, des méthodes furent développés pour palier aux difficultés d'estimation des polynômes de la définition originale. Dans son article de 2001, Alexa définit la surface de projection MLS ou "PSS" d'un nuage de points comme l'ensemble des points stationnaires par l'opérateur de projection [Alexa et al., 2001]. Une des méthodes pour évaluer cette surface estime itérativement la position du point x en projetant celui-ci sur un hyperplan moyen. Amanta, notamment dans [Amanta and Kil, 2004], prouve qu'il suffit d'approcher la surface par un polynôme de degré 1, c'est-à-dire le plan, et qu'en itérant cette procédure jusqu'à stabilisation du point on obtient sensiblement la même position finale du point x . Ainsi la procédure mise en place dans notre chaîne de traitement va itérer les procédures suivantes :

- dans un premier temps l'hyperplan local H_x correspondant aux données du voisinage pondéré $V(x)$ d'un point x est estimé,
- puis dans un deuxième temps le point 3D x est projeté sur ce plan.

Dans cette méthode, il existe deux points clefs que nous tenterons de mettre en lumière. Le premier point est le calcul du voisinage pondéré. En effet, la position de l'hyperplan dépendra, comme nous le verrons dans la suite, de la position du centroïde. Or la position de ce centroïde dépend intégralement de la pondération du voisinage calculé par un noyau de pondération. Ainsi le choix du noyau de pondération est un facteur crucial dans cette méthode.

Le deuxième point est la définition de projection. En effet, la manière de projeter le point sur l'hyperplan changera radicalement le résultat obtenu.

Approximation de l'hyperplan

Pour approcher l'hyperplan correspondant au voisinage, il est nécessaire de connaître le voisinage. Celui-ci sera trouvé par une procédure de recherche des k -plus proches voisins, telle qu'elle est expliquée dans la section 2.1.1. Le problème d'utiliser une technique de lissage est que nous pouvons perdre des détails, notamment sur les bords d'objets où les angles seront rabotés. De plus dans le cas où de nombreuses surfaces coexisteraient, comme c'est le cas dans nos images de végétation, les surfaces auront tendances à s'étirer sur les bords des surfaces.

Une fois le voisinage trouvé, il est dit dans [Alexa et al., 2001] et [Adamson and Alexa, 2003], que le meilleur plan est celui qui minimise localement la somme des carrés des distances pondérées du voisinage au plan :

$$H_x = \operatorname{argmin} \sum_{i=1}^N (\langle n, p_i \rangle - x')^2 \theta(d(p_i, x')) \quad (2.5)$$

Algorithm 3 Surface MLS par projection itérative sur un plan :

```

while  $|x - x'| > \lambda$  do
    calcul de  $c(x)$  et  $n(x)$ 
    projection de  $x$  sur  $H_x$ 
end while

```

avec θ noyau de pondération, p_i les points du voisinage de x et x' le projeté de x sur H_x et d la distance Euclidienne et n le vecteur normal. Pour le noyau de pondération, de nombreuses possibilités ont été développées, généralement basées sur un noyau de pondération Gaussien défini tel que :

$$\theta(d) = e^{-\frac{d^2}{h^2}}$$

Avec h un facteur de régularisation. Ce facteur est très important, car on peut le choisir constant pour tout l'objet ou en fonction de la densité locale des points, il faut savoir que ce facteur peut avoir un effet très lissant si les objets ont des tailles inférieures à h . Une autre possibilité est de remplacer le noyau de pondération par une fonction à support compacte.

Dans l'état de l'art, la manière première de résoudre l'équation 2.5 s'effectue en plaçant $x' = x + \beta n$ et par une minimisation itérative non-linéaire. Mais, le minimum local est trouvé avec la valeur minimal de β , donc H_x est alors le plan tangent du voisinage le plus proche de x comme expliqué dans [Alexa and Adamson, 2004], on évite ainsi cette minimisation.

La normale n de x dans le champ des normales à la surface sera approchée par la moyenne pondérée des normales des voisins si elles sont fournies. Ce qui n'est pas le cas dans notre problématique, la direction du point à la caméra sera alors choisie comme approximation de la normale. La normale n peut être calculé comme :

$$n(x) = \frac{\sum_i \theta(d(x, p_i)) n_i}{\sum_i \theta(d(x, p_i))} \quad (2.6)$$

Avec d la distance euclidienne.

Pour une meilleure valeur de la normale, un calcul du plus grand vecteur propre de la matrice de covariance du jacobien de H_x peut être envisagé, mais pour des raisons de coûts temporels et combinatoire, l'approximation des normales a été privilégiée.

On définit le centroïde pondéré des données $c(x)$ définit comme 2.9 :

$$c(x) = \frac{\sum_i \theta(d(x, p_i)) p_i}{\sum_i \theta(d(x, p_i))} \quad (2.7)$$

On peut résumer l'algorithme utilisé dans l'algorithme 3.

avec λ un scalaire de contrôle, avec $\lambda = 0$ dans la cas d'une surface parfaite avec une stabilisation complète du point. En réalité, on choisit d'itérer au moins un certain nombre k de fois et si $|x^k - x^{k+1}| < 0.1|x^{k-1} - x^k|$ alors l'algorithme s'arrête.

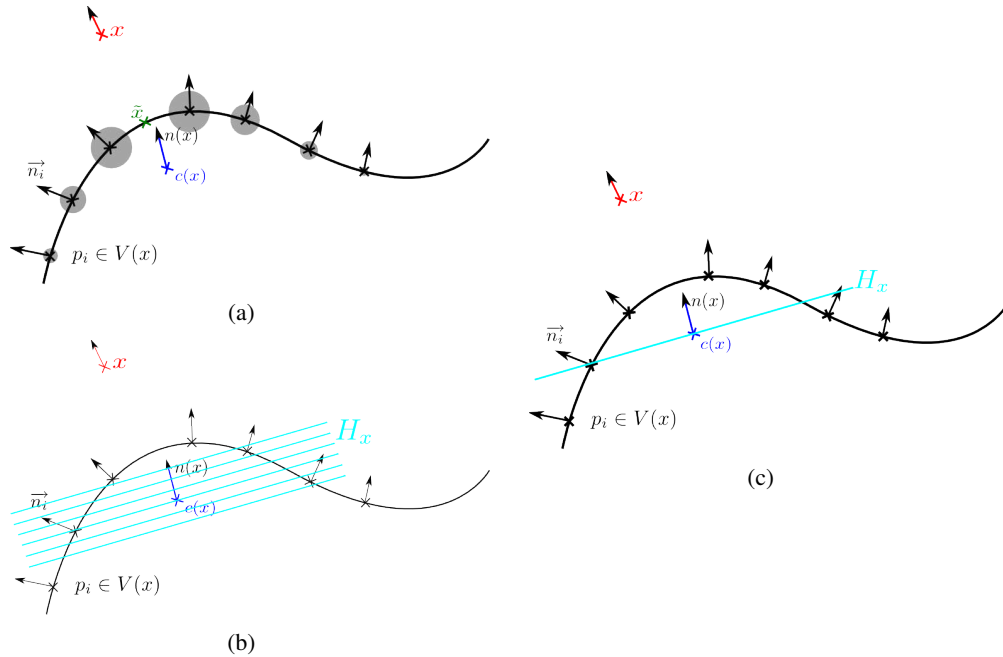


FIGURE 2.3.1 – (a) Situation originale du point bruité x et \bar{x} sa position parfaite, dans un cas non bruité. Les cercles gris représentent les valeurs des pondérations du voisinage (b) Une fois $c(x)$ et $n(x)$ estimés, une ensemble de plan H_x est défini selon $n(x)$ et en (c) le plan H_x pour la plus petite valeur de β .

Le seul point critique de l'approximation de H_x repose sur le noyau de pondération. En effet en fonction de celui-ci le plan sera grandement différent. De nombreuses recherches ont été faites pour trouver le meilleur noyau possible. Mais la simplicité du noyau Gaussien simple et son caractère plus lissant dans des cas très bruités, ce qui est notre cas, nous a fait préférer cette approche simple. L'état de l'art nous apprend que le résultat sur nos nuages serait sensiblement équivalent, voir meilleur, ceci lié à la présence dans nos nuages de trop de bruits.

Une variante de cet algorithme a été développée et propose la projection de x non plus sur un plan mais sur une sphère algébrique [Guennebaud and Gross, 2007], car il est montré que le plan est un cas dégénéré de la sphère algébrique et que celle-ci approche beaucoup mieux les données surtout dans le cas de nuages avec des taux d'échantillonnage faibles par rapport à la courbure, ce qui n'est pas notre cas. Ces rares cas dégénérés que peut produire la procédure ci-dessus disparaissent quand la projection se fait sur la sphère algébrique. Pour des raisons de temps, cette approche prometteuse n'a pas été testée sur nos jeux de données.

projection

Une fois le plan de projection approché, il reste à étudier les procédures de projection de x sur H_x . Or cette étape de projection est un point clef de l'algorithme. L'état de l'art nous apprend qu'il existe trois types bien définis de projection

[Alexa and Adamson, 2004] : basique, quasi-orthogonale et orthogonale. Une fois le centroïde $c(x)$ et la normale locale $n(x)$ calculés, seul l'emplacement du projeté changera. Pour un point x , son projeté sur l'hyperplan H_x approché à l'étape précédente est x' .

Ces trois procédures de projection peuvent être définies comme :

- Basique : $x' = x - n(x)^T(x - c(x))n(x)$ comme vu sur la figure 2.3.2. C'est-à-dire, qu'à chaque itération, le point est reprojété orthogonalement sur le nouveau plan calculé.

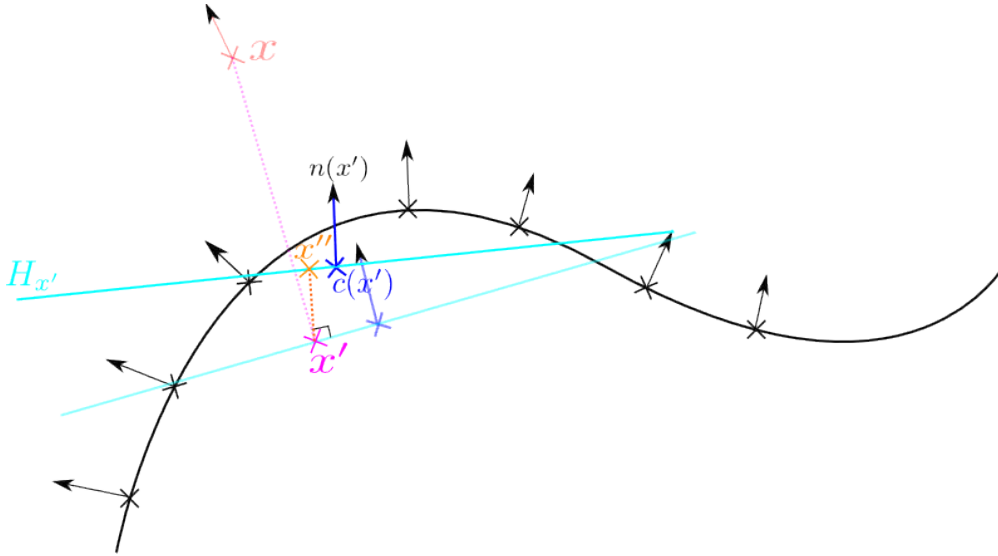


FIGURE 2.3.2 – Ici on peut voir que le point x se projette selon $n(x)$ sur H_x en x' , ici marqué en violet, ensuite une nouvelle itération démarre, $H_{x'}$ est approché et x' est projeté selon $n(x')$ sur $H_{x'}$ en x'' ici en orange.

- quasi-orthogonale : en considérant $x_0 = x$ avant les itérations, on projette le point original dans le nouveau système $x' = x_0 - n(x)^T(x_0 - c(x))n(x)$ comme vu sur la figure 2.3.3. Ce qui implique que c'est toujours le point original qui est reprojété, même si c'est le projeté qui interviendra dans le calcul du nouveau plan de projection.
- orthogonale : en considérant $x_0 = x$ avant les itérations et $f(x) = n(x)^T(x - c(x))$ correspondant à la fonctionnelle approchant les points par rapport à l'hyperplan H , son gradient $\nabla f(x)$ décrivant les normales à la surface, la projection se fait alors comme $x' = x_0 - \frac{n(x)^T(x - c(x))}{n(x)^T \nabla f(x)} \nabla f(x)$ comme vu sur la figure 2.3.4. Ainsi le point sera parfaitement projeté selon $\nabla f(x)$ et non plus selon la moyenne des normales.

Une fois la projection effectuée la procédure est itérée jusqu'à stabilisation de la position du point.

2.3.4 Surface de projection colorimétrique

Une différence majeure existe entre les nuages de points produits par des scanners 3D et ceux produits par notre chaîne. C'est la présence dans nos nuages d'informations

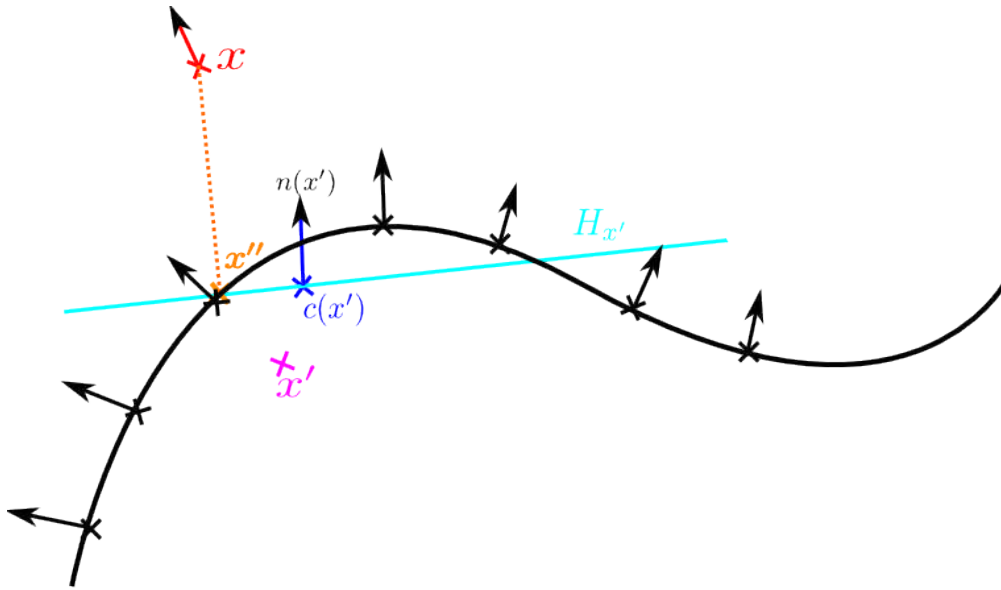


FIGURE 2.3.3 – Ici on peut voir que le point x se projette selon $n(x)$ sur H_x en x' , ici marqué en violet, ensuite une nouvelle itération démarre $H_{x'}$ est estimé et x est projeté selon $n(x')$ sur $H_{x'}$ en x'' ici en orange

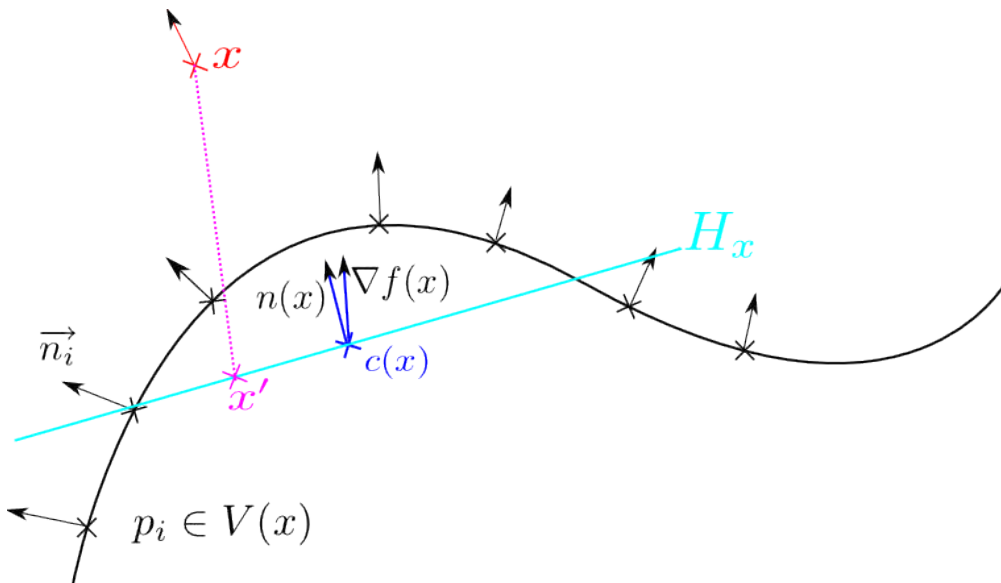


FIGURE 2.3.4 – Dans le cas d'une projection orthogonale, x est projeté sur H_x selon $\nabla f(x)$ en x' et non plus selon la moyenne normalisée pondérée des normales des voisins.

colorimétriques. Nos nuages représentent des scènes réelles, elles comportent les informations colorimétriques des images dont elles sont issues.

Nous avons choisi d'ajouter cette information au noyau de pondération du voisinage. Le principal intérêt de cette méthode est qu'elle respecte mieux la géométrie et ne

lisse que dans les zones de couleurs homogènes. Ce phénomène vient du fait que dans une scène naturelle les objets différents ont généralement une couleur différente. Nous avons donc rajouter une pondération colorimétrique au voisinage. Cette pondération supplémentaire implique qu'un point voisin prendra en compte la distance spatiale et colorimétrique.

Malheureusement, utiliser une telle contrainte, ne nous permet plus de justifier de la propriété de continuité des surfaces MLS comme nous pouvons le voir dans le cas des ombres dans l'image 2.3.5.

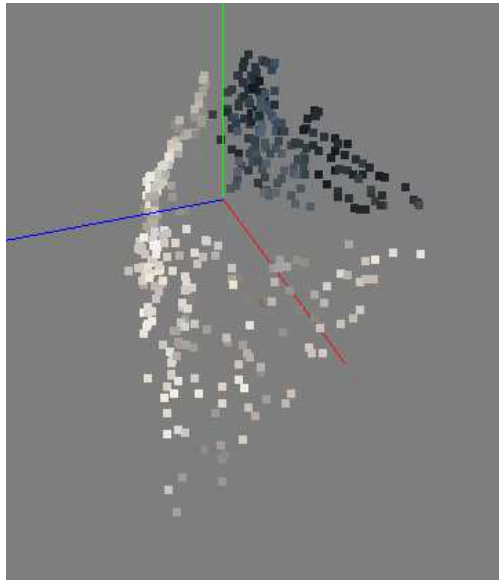


FIGURE 2.3.5 – Ici on peut voir que l'ombre d'une pierre, qui devrait appartenir à la même surface que le roc en beige, présente une discontinuité avec celle-ci.

Pour éviter ce problème avec l'ombre, nous avons choisi l'espace colorimétrique Lab, permettant de minimiser l'influence de la luminance dans la distance couleur. L'espace colorimétrique Lab est différent de celui RGB classique car il prend mieux en compte la perception des couleurs, ainsi L représente la luminance de la scène quand a et b représentent respectivement la gamme de l'axe rouge à vert en passant par le gris et la gamme de l'axe jaune à bleu en passant par le gris. En minimisant l'importance de la luminance dans la distance, on évite le problème avec les ombres.

Pour intégrer cette pondération, la recherche des centroïdes et normales moyennes est modifiée telle que :

$$c(x) = \frac{\sum_i \theta(d(x, p_i)) \zeta(d_{lab}(x, p_i)) p_i}{\sum_i \theta(d(x, p_i)) \zeta(d_{lab}(x, p_i))} \quad (2.8)$$

$$n(x) = \frac{\sum_i \theta(d(x, p_i)) \zeta(d_{lab}(x, p_i)) n_i}{\sum_i \theta(d(x, p_i)) \zeta(d_{lab}(x, p_i))} \quad (2.9)$$

Avec ζ le noyau de pondération colorimétrique qui, pour des raisons de simplicité calculatoire, est une pondération binaire utilisant un seuillage et d_{lab} la distance colorimétrique dans l'espace Lab.

Une discontinuité persiste mais comme elle est faible et comme le choix d'un voisinage colorimétrique empêche un lissage trop important entre des objets différents, nous avons choisi de garder cette approche.

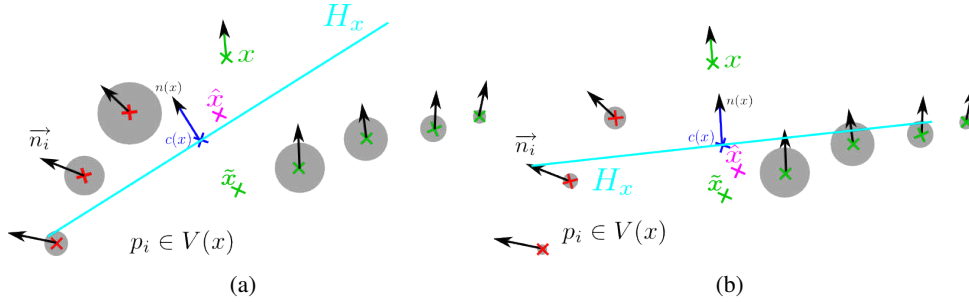


FIGURE 2.3.6 – Schéma de l'algorithme PSS en (a) avec le problème visible de normalisation des surfaces, alors qu'en (b) l'algorithme CPSS montre que le poids des points dépendant de la couleur, la normalisation est moins forte.

L'intérêt de CPSS réside aussi dans le fait que comme les caractéristiques de la scène sont mieux conservées, le nombre d'itération peu être plus fort car le phénomène de normalisation est beaucoup moins important. Évidemment la limitation viens du fait que si deux surfaces ont parfaitement la même couleur, alors les problèmes seront les mêmes qu'avec l'algorithme PSS. Il nous semblerait intéressant de tenter de rajouter ce noyaux à d'autres algorithmes plus complexes, malheureusement un manque de temps de nous a pas permis de tester avec robustesse ces approches.

2.3.5 Résultats et comparaisons

Le gros problème de nos nuages est la présence récurrente de bruit, qui peut parfois être très important. Ainsi les méthodes les plus robustes ont été testées, les "Robuste Implicite MLS" ou "RIMLS" ainsi que les "Algebraic Point Set Surface" ou "APSS", définie dans [Guennebaud and Gross, 2007], pour les techniques les plus récentes. Nous avons aussi testé la méthode, plus ancienne d'adamson [Adamson and Alexa, 2003] nommé PSS.

Le problème des RIMLS, et dans une moindre mesure de l'APSS, est que la densité des points est trop forte. Dans notre cas la densité de point est très forte ce

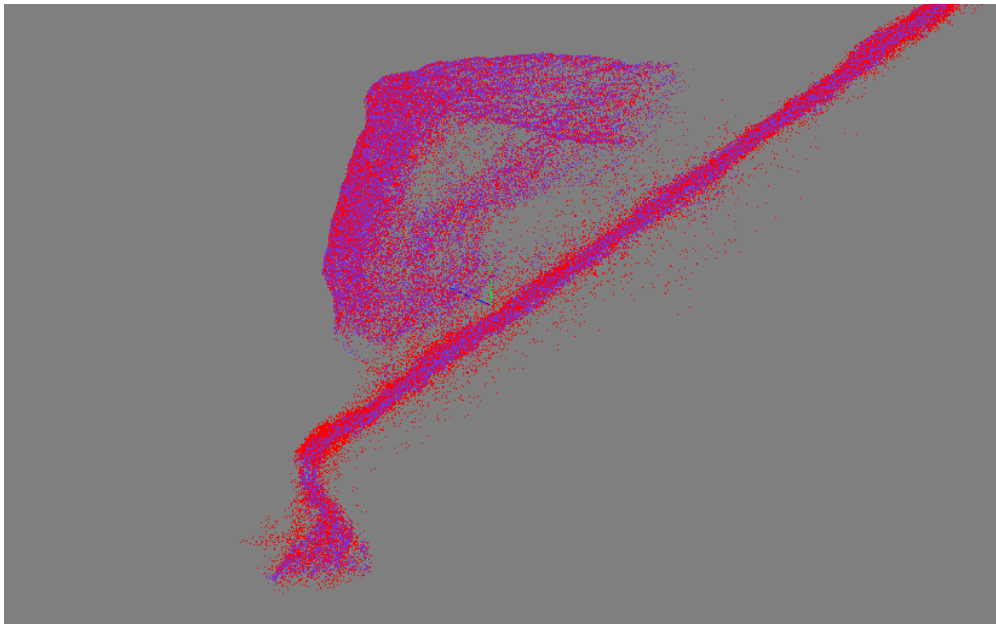


FIGURE 2.3.7 – Différence entre PSS, en rouge, et CPSS en violet ; montrant notamment dans la partie inférieur gauche que la marche est mieux gardée que dans le nuage rouge.

qui tend à expliquer pourquoi à contrario la méthode PSS marche si bien. Elle a un caractère beaucoup plus lissant et ne tente pas de préserver les objets, ceux-ci seront conservés par le choix du voisinage colorimétrique, qui empêche l'effet de normalisation du lissage. Ainsi utiliser dans notre cas des techniques RIMLS ou APSS qui eux utilisent des algorithmes plus complexes pour la conservation de ces détails dans un cas peu dense, seraient inutiles et ralentiraient le traitement des données. La comparaison des résultats peut être vue sur la figure 2.3.8

Nous pouvons voir les résultats pour nos objets avec la méthode 2.3.9, 2.3.10, 2.3.11 et 2.3.12.

Conclusion

L'utilisation de la chaîne de traitement expliquée dans le premier chapitre nous a imposé de créer une chaîne complète de traitement 3D visant à obtenir le meilleur nuage possible. Nous nous sommes aperçus que peu de littérature existaient pour nos problèmes, nous obligeant à créer des algorithmes ou détourner de leur utilisation première certains algorithmes qui ont donné satisfaction.

Nous avons pu notamment utiliser le voisinage pour supprimer les points aberrants, vérifiant ainsi notre hypothèse sur la corrélation entre position aberrante et nombre de voisins. De plus, la création complète de l'algorithme visant à supprimer les redondances nous a permis une manipulation simplifiée du nuage. En effet, nous avons fait plusieurs tests de chaîne pour voir dans quel ordre les algorithmes devaient

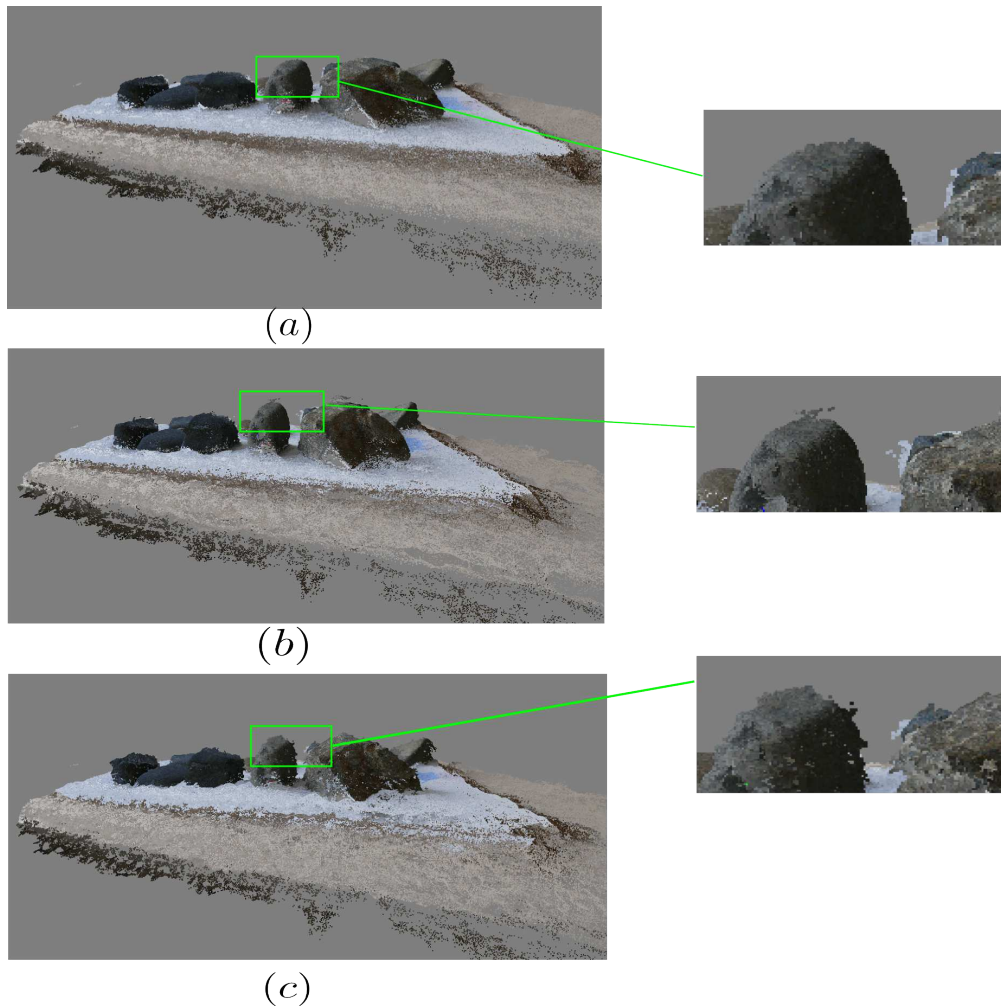


FIGURE 2.3.8 – (a) Résultat obtenu avec la méthode PSS (b) Résultat obtenu avec la méthode des RIMLS (c) Résultat obtenu avec la méthode de l'APSS.

être effectués.

Si la suppression des points aberrants nous est apparue comme étant logiquement la première étape à effectuer, il semblait plus logique de lisser les nuages avant de supprimer les redondances. Or, même si elle n'est pas primordiale, la contrainte de temps nous a empêché de le faire car le lissage par CPSS est assez lent (même si une parallélisation accélérerait le processus). Ainsi, la chaîne de traitement, qui prend dans le sens présenté une dizaine de minutes à être effectuée sur une vingtaine de nuages de trois millions de points, prend dans notre test plus de 6 heures pour un résultat inférieur. Ces observations nous ont confortées dans notre choix de garder la chaîne dans cet ordre.

Ainsi, les résultats concluants obtenus nous ont permis de poursuivre et de pouvoir effectuer des mesures sur ces nuages. Mais, avant d'étudier les mesures, il est utile d'étudier leur robustesse. Or, la confiance dans les mesures est dépendante de la

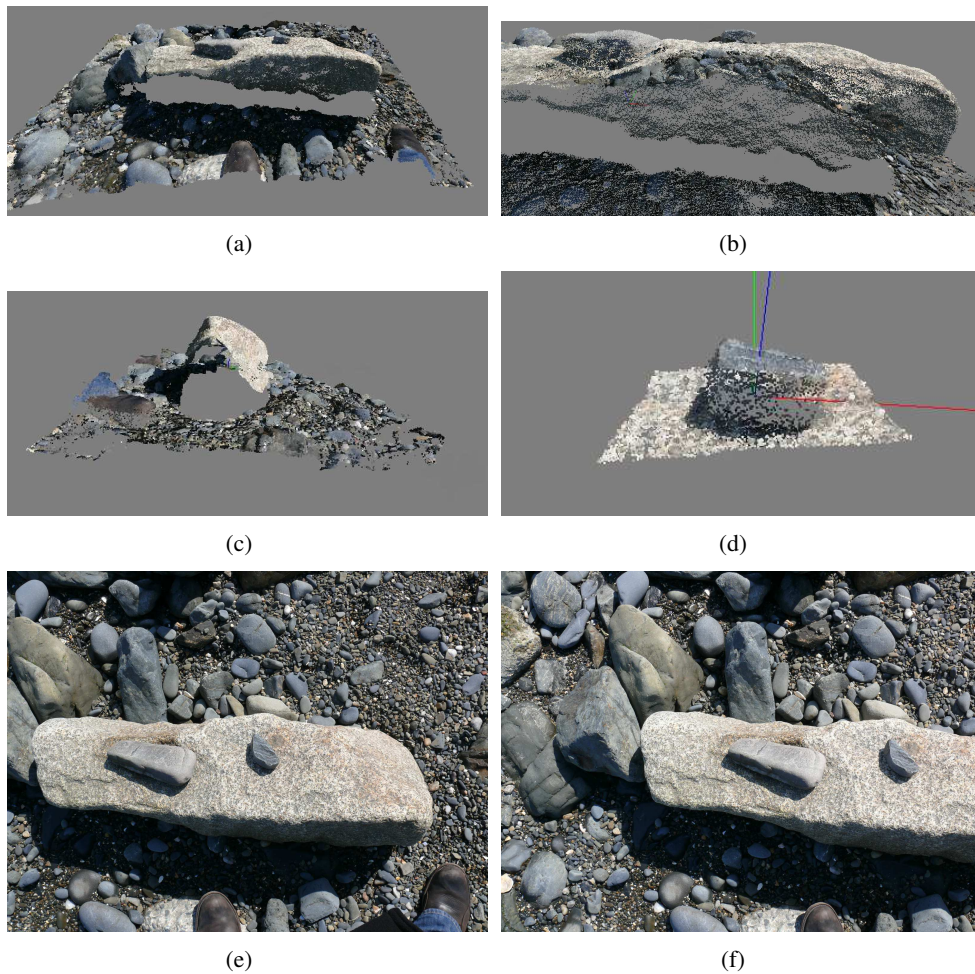


FIGURE 2.3.9 – (a)(b)(c) Vue de coté de la reconstruction du nuage du rocher. (d) Gros plan sur une partie du nuage sur lequel on peut observer l'effet de lissage souhaité pour le calcul de mesures sur le nuage. (e)(f) Exemples des 19 images.

confiance que l'on accorde à la reconstruction 3D obtenue. Ainsi, il nous paraissait essentiel d'effectuer une campagne de mesures de tests pour vérifier la véracité des mesures. Nous verrons que pour des raisons de temps nous n'avons pas pu effectuer une campagne de véritables tests. Nous avons donc fait le choix de vérifier les mesures sur un jeu test de photos dont nous connaissons les mesures. Nous verrons que cela nous permet de donner un ordre de grandeur à la robustesse des algorithmes utilisés.

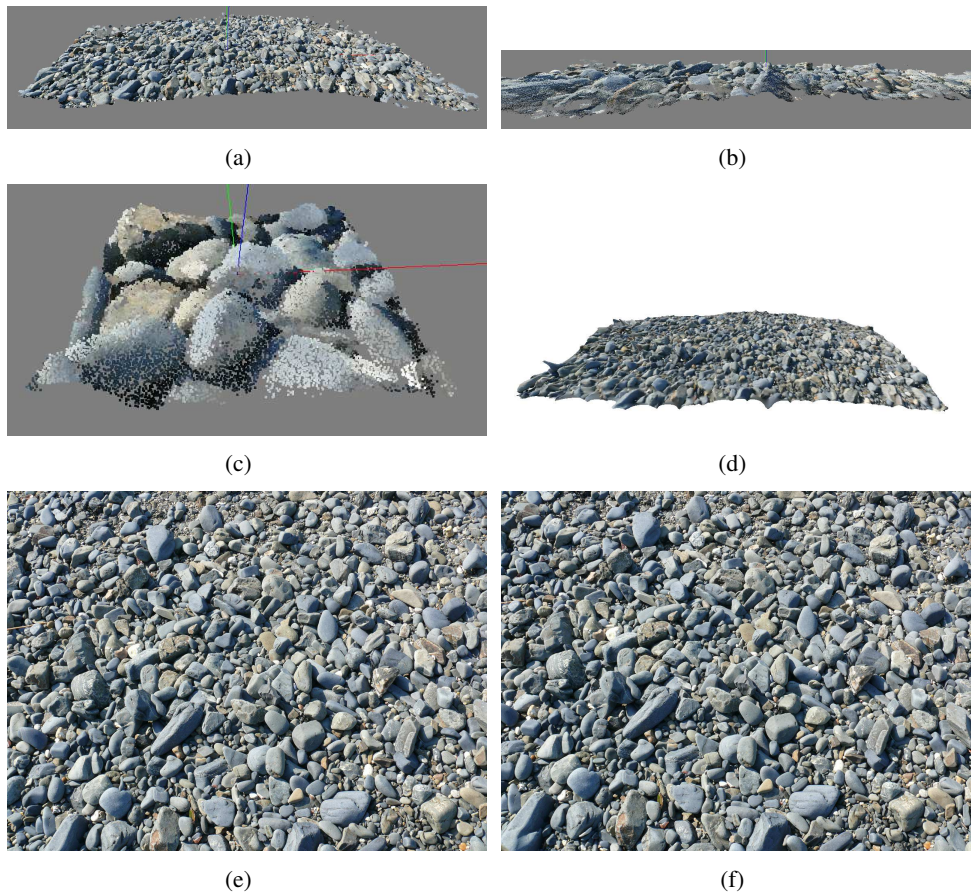
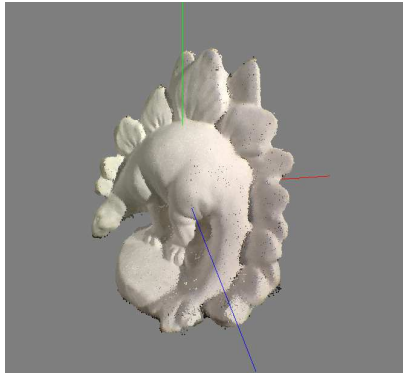
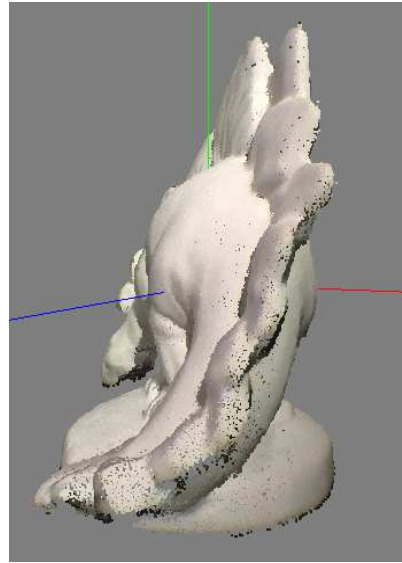


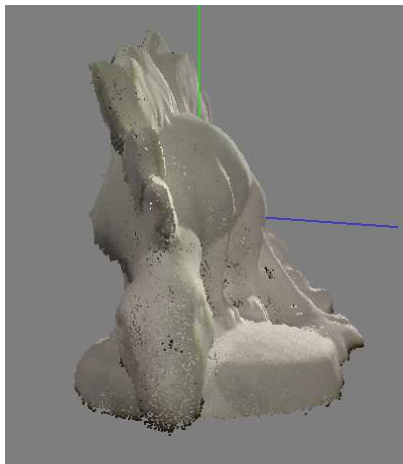
FIGURE 2.3.10 – (a)(b)Vues du nuage de galets. (c) Gros plan du nuage. (d) Reconstruction avec maillage 3D, inutile pour des mesures mais permet de voir la qualité de la reconstruction.(e)(f) Exemples des 35 images.



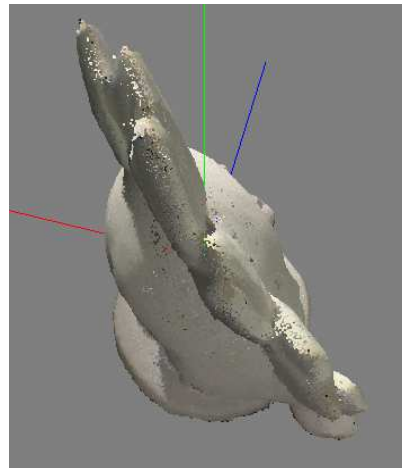
(a)



(b)



(c)



(d)



(e)



(f)

FIGURE 2.3.11 – (a)(b)(c)(d) Vue de coté de la reconstruction du dinosaure. (e)(f) Exemples des 48 images.

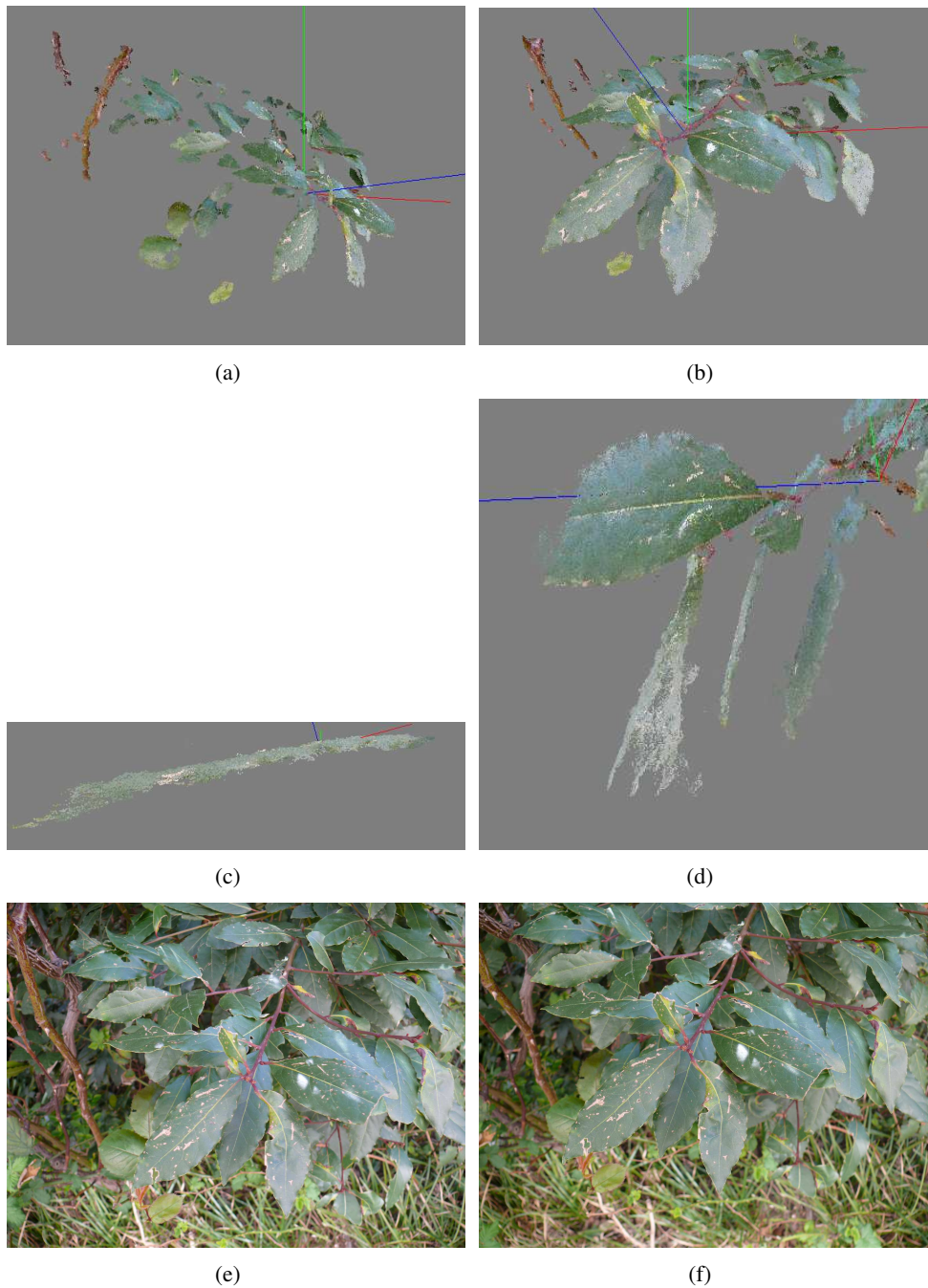


FIGURE 2.3.12 – (a)(b) Vue du nuage final de points. (c) Exemple sur une feuille, on peut voir l’affinement de la surface liée aux caractéristiques de la projection MLS. (d) Exemple d’un problème lié à la reconstruction de feuilles. Il est très difficile de filtrer ce problème car la feuille est bien reconstruite et on ne peut pas parler d’outliers. Alors ni l’ACC, ni le filtrage colorimétrique ne peuvent supprimer cet effet, or sur l’exemple, l’algorithme ne trouve pas un surface pour la feuille mais une surface pour chaque position de la feuille, voilà pourquoi elle apparaît triple.(e)(f) Exemples des 22 images.

Chapitre 3

Résultat, validation et discussion

Avant d'aborder l'étude des paramètres de surfaces extractibles sur nos nuages, il convient d'abord de vérifier la cohérence des mesures effectuées. Pour cela, nous aurions du nous comparer avec une réalité terrain extrêmement difficile à trouver et représentant un temps très important pour pouvoir affirmer la véracité des mesures trouvées. Nous avons donc fait un choix, celui de créer un nuage de points 3D d'une scène dont nous connaissons les mesures et de les comparer à des mesures faites sur le nuage. Cela n'a pas pour but de mesurer l'exactitude de la reconstruction mais de donner un ordre d'idée de la qualité globale de la méthode.

Nous avons fait le choix de faire la chaîne de traitement la plus générique possible. Pour montrer la robustesse et la généralité de la chaîne de traitement, nous avons choisi d'appliquer nos algorithmes à d'autres types de scènes que celles en parfaite adéquation avec notre sujet. Avoir cette information 3D nous permet de faire toutes les hypothèses de mesures futures, ce travail n'est donc pas vain et prouve la pérennité du travail effectué.

Dans ce chapitre, nous aborderons trois points différents :

- la généralité de la méthode s'adaptant à tout type de scènes ;
- la validation des résultats. Ceux-ci ont été testés sur un banc de test prévu pour ressembler à une scène naturelle sur laquelle l'ensemble des paramètres peut être contrôlé. Nous avons fait le choix d'utiliser de vraies images et non simulées, nous expliquerons dans la partie réservée au banc de test les raisons de ce choix ;
- une discussion sur la provenance des erreurs restantes sur les mesures. Trois hypothèses seront abordées :
 - la scène à reconstruire et la manière dont les photos ont été prises ;
 - les algorithmes choisis (auto-calibrage et mise en correspondance) ;
 - la manière de mesurer ;

Cette discussion à propos des erreurs aboutira à une réflexion sur la manière d'obtenir le meilleur résultat possible. Ce résultat optimum sera trouvé sous certaines conditions de prises de vues. Ainsi, nous donnerons une suite de conseils sur la manière de faire les prises de vues de scènes à reconstruire.

3.1 Résultats

Ne connaissant pas les utilisations possibles des nuages de points, il semblait normal de tester la chaîne de traitements sur d'autres types de données que celles pour l'obtention des paramètres bio-géo-physiques. Nous testerons donc notre méthode sur des photos représentant des scènes très diverses.

Ces données ont été choisies car elles correspondent à des problèmes courants en reconstruction 3D. Ainsi, nous avons choisi de tester nos chaînes sur des images de bâtiments, d'objets d'art et de visages. Il existe de nombreux algorithmes spécialisés dans l'une ou l'autre de ces données et nous avons voulu tester ainsi la généralité de la chaîne de traitements. Notre méthode n'a pas vocation à concurrencer ces algorithmes dans chaque domaine mais plutôt de montrer sa robustesse aux différents types de scènes à reconstruire. Nous nous contenterons de montrer pour chaque section des résultats et d'en faire le commentaire, l'évaluation des résultats étant uniquement visuelle. Nous concluons cette partie en nous comparant à d'autres chaînes complètes de traitements.

3.1.1 Bâtiments

La reconstruction de bâtiment est un point clef de la stéréo-vision. Généralement celle-ci se fait par imagerie aérienne ou satellitaire pour une reconstruction rapide de ville. L'Institut National Géographique participe activement à la mise en place d'une cartographie 3D des villes françaises. En plus des images aériennes, des projets plus importants ont vu le jour pour reconstruire les façades de bâtiments (TerraNumérica). Le problème de ces méthodes est qu'un équipement important et cher est nécessaire. Et même si le résultat est à la hauteur des moyens mis en œuvre, il reste hors de portée des particuliers ou de chercheurs avec des moyens limités. Or, notre méthode à bas coût, permet de reconstruire les façades avec une bonne évaluation visuelle, même si elle est de moins bonne qualité.

3.1.2 Objets

La problématique de reconstruire des objets d'art ou de patrimoine est très éloignée de notre problématique. Mais des mesures effectuées sur ce type de données pourraient être intéressantes. La bibliographie nous apprend que pour reconstruire ces objets, deux approches ont été mises en place :

- l'utilisation d'un scanner laser, comme dans le cas de la reconstruction du David de Michel-Ange [Koller et al., 2004].
- l'utilisation d'un plateau tournant et d'images calibrées par une mire [Scharstein and Szeliski, 2002], [Hernández Esteban and Schmitt, 2004].

Dans les deux cas, les prises de vues sont complexes et nécessitent beaucoup de temps et de moyens. Nous ne souhaitons pas appliquer ces contraintes dans notre chaîne. Nous tenterons donc de voir les résultats obtenus par notre méthode sur des objets d'art pour lesquels le calibrage est donnée.

Certains objets ont été reconstruits à partir d'images fournies dans le cadre d'un classement. Dans ce cas, un fichier de calibrage accompagnait les images. La



FIGURE 3.1.1 – Reconstruction d’un bâtiment avec notre chaîne de traitements à partir de 61 images. Les images de ce bâtiment données en exemple nous sont fournies par M.Furukawa.

méthode de prise de vue consiste à prendre des photos d’un objet, placé sur un plateau tournant. Plusieurs jeux d’images sont disponibles : 8, 48 et 312 images. Nous avons

choisi de travailler sur le cas de 48 images, le cas de 8 images n'offrant pas assez de recouvrement et le cas 312 n'offrant que peu de différence avec le cas 48. Notre méthode produisant des nuages de points denses, utiliser 312 images ne fait que renforcer cette densité, sans offrir une véritable amélioration visuelle.

Nous n'avons pas testé nos résultats dans le cadre du classement, car il est pratiqué sur une surface et non des points, or n'ayant pas créé de surface nous ne pouvions pas tester notre algorithme.

Aucun algorithme n'a été spécialisé, le but étant simplement de montrer la qualité visuelle des reconstructions. Les fichiers contenant les paramètres de calibrage fournis par le classement ont été utilisés car les EXIFs manquaient dans les images. Il ne s'agit donc pas parfaitement de notre chaîne de traitement. Mais, la généralité de la chaîne est prouvée par la possibilité d'utiliser différentes possibilités pour chaque maillon de la chaîne. Deux exemples ont été testés sur les figures 2.3.11 et 3.1.2, où est reconstruite une statue de dinosaure et une maquette d'un temple grec.

3.1.3 Visages

L'actualité nous a prouvé l'intérêt de reconstruire en 3D des visages, avec l'utilisation d'acquisition de visages en temps réel pour les jeux vidéos ou le cinéma. Il est notable de voir un véritable engouement pour les reconstructions de visages en 3D pour l'immersion dans des mondes imaginaires. De nombreuses recherches existent sur le sujet. Comme dans le cas des objets d'art, reconstruire un visage nécessite un équipement d'acquisition très complexe. En effet, le but est d'obtenir en même temps différents angles de vues du visage. Cette manière de capturer les visages est loin de notre problématique. Dans notre cas, il est nécessaire que le sujet reste le plus immobile possible le temps de prendre les différentes prises de vues.

La difficulté de travailler sur des visages réside dans l'objet lui-même, c'est-à-dire la personne prise en photo. Celle-ci doit rester immobile le temps de la prise des photos, ne pas avoir de réflexion spéculaire sur la peau, liée notamment à la présence de sébum, et surtout ne doit pas suivre l'objectif des yeux. Les yeux humains ne pouvant rester immobiles, les sujets sont généralement contraints de fermer les yeux. Nous pouvons voir ci-dessous les résultats obtenus sur un collègue, Tamy Boubekeur.

3.1.4 Comparaisons

Il est important de tester d'autres chaînes de traitement pour nous comparer. Ces chaînes optent généralement entre deux optiques de reconstruction différentes :

- Une optique de l'image à la 3D. C'est le cas de notre méthode.
- Une optique de la 3D à l'image. C'est le cas des méthodes avec lesquelles nous voulons nous comparer.

Dans les méthodes utilisant la 3D comme point de départ, il est possible de voir deux idées directrices se dégager :

- Une minimisation d'énergie sur un espace discrétisé ;
- Une méthode de grossissement de plans ;



FIGURE 3.1.2 – Vue de la reconstruction du temple.

Les méthodes utilisant une discrétisation de l'espace, considère l'espace 3D comme un graphe 3D dans lequel chaque nœud de cet espace est reprojété dans les images pour calculer une énergie. Il devient alors possible de trouver les



FIGURE 3.1.3 – Vue du visage reconstruit du D.Tamy Boubekeur.

nœuds correspondants à la surface 3D. La technique la plus connue utilisant ce type d'approche cherche la surface de coupure minimum du graphe [Roy and Cox, 1998] par la technique dite des "graph cut" [Boykov et al., 1998]. Cette technique offre de bons résultats mais malheureusement n'a pu être appliquée à nos images car la taille du graph est trop grande pour le niveau de précision souhaité. En effet, de la discrétisation de l'espace dépend la résolution 3D, or dans notre cas, la mémoire nécessaire est trop importante pour un PC du commerce ¹. Ce problème ne saurait être

1. 4Go de RAM dans le PC utilisé

une limitation à l'utilisation de telle méthode.

La principale limitation nous empêchant d'utiliser ces approches est la nécessité d'avoir un calibrage très précis, or ce n'est pas notre cas. Les techniques utilisant une discrétisation de l'espace 3D ne sont donc pas envisageables.

Dans le cas d'algorithme utilisant des méthodes de grossissement de plan, l'algorithme développé par Furukawa en 2009 [Furukawa and Ponce, 2009], nous est apparu comme une très bonne alternative car il n'utilise pas la condition de calibrage parfait. Cet algorithme, classé parmi les meilleurs algorithmes du classement de Middlebury [Scharstein and Szeliski, 2002], a montré, sur les tests réalisés en 3.1.5, un bon comportement dans la plupart des cas.

Le principe de cette technique est de faire croître des plans à partir de graines. Des points SIFT sont généralement utilisés comme graine. Cette technique a montré une limitation dans le cas d'images comportant une trop grande succession de plans, comme dans le cas de la végétation (voir figure 3.1.6). Ce phénomène est problématique car les mesures sur la végétation sont un des points clefs de cette thèse, aussi cette méthode ne peut être choisie. On peut tout de même observer le très bon comportement de l'algorithme dans le cas des bâtiments, pour cela nous pouvons comparer notre méthode 3.1.1 à celle-ci 3.1.4.



FIGURE 3.1.4 – Reconstruction 3D obtenue par l'utilisation de la chaîne de traitement de Furukawa avec les mêmes paramètres de calibrage que dans notre reconstruction 3.1.1.

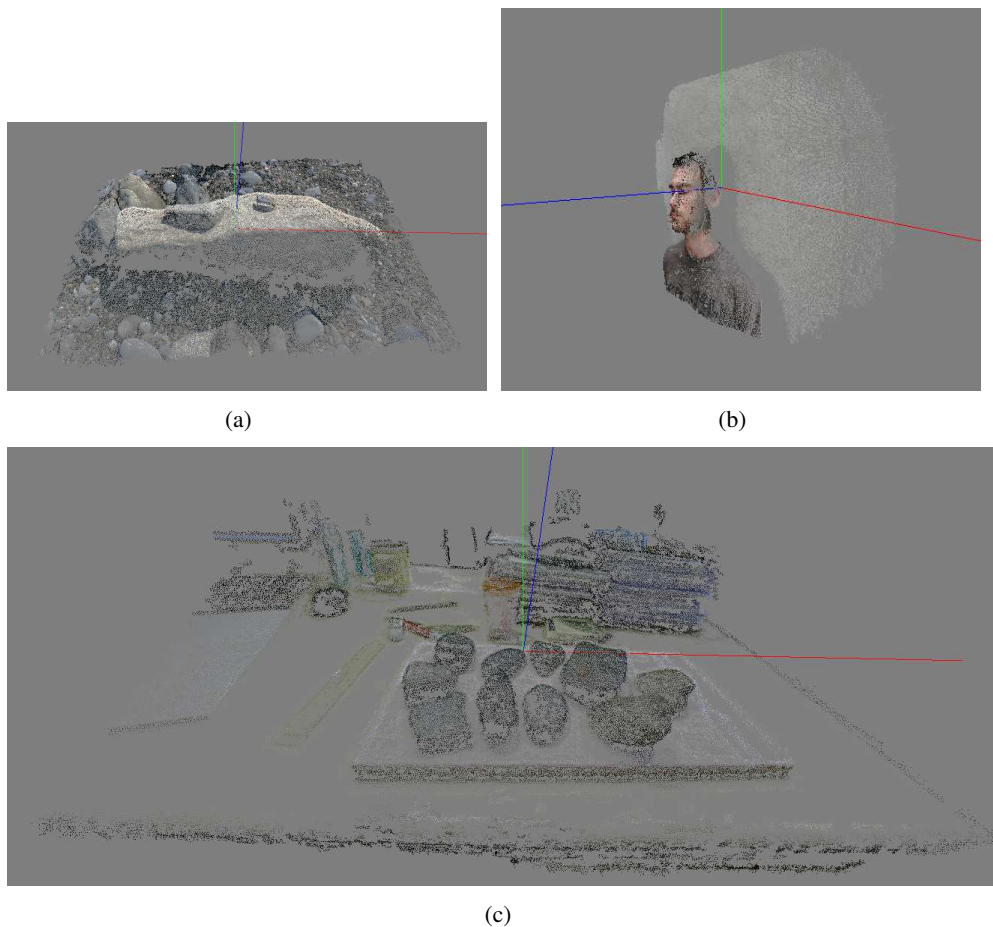


FIGURE 3.1.5 – Exemples de reconstructions obtenues par l'utilisation de la chaîne de traitement de Furukawa. Les résultats sont comparables voire meilleurs visuellement, malgré une moins grande densité de points.

Il est notable de remarquer sur la figure 3.1.6, que l'algorithme de Furukawa a deux problèmes majeurs qui sont un frein à leur utilisation pour l'extraction de paramètres biologique et géo-physiques. Le premier défaut est que l'algorithme n'est pas robuste aux mouvements des végétaux. Le deuxième défaut est le manque de densité des nuages de points reconstruits. L'algorithme cherche des plans à faire grossir et excelle visuellement dans la reconstruction de plans comme dans le cas de bâtiments, il est donc normal que l'algorithme peine sur les végétaux.

3.2 Validation

Des mesures peuvent être effectuées sur le nuage 3D, mais celles-ci seront fiables quand elles auront été vérifiées. En effet, pour juger la fiabilité des mesures sur les nuages, il est nécessaire de connaître des correspondances entre mesures réelles et celles réalisées sur les nuages 3D. Il faut donc avoir un outil capable de faire des mesures sur le nuage de points 3D (outil que nous nous sommes fourni au travers

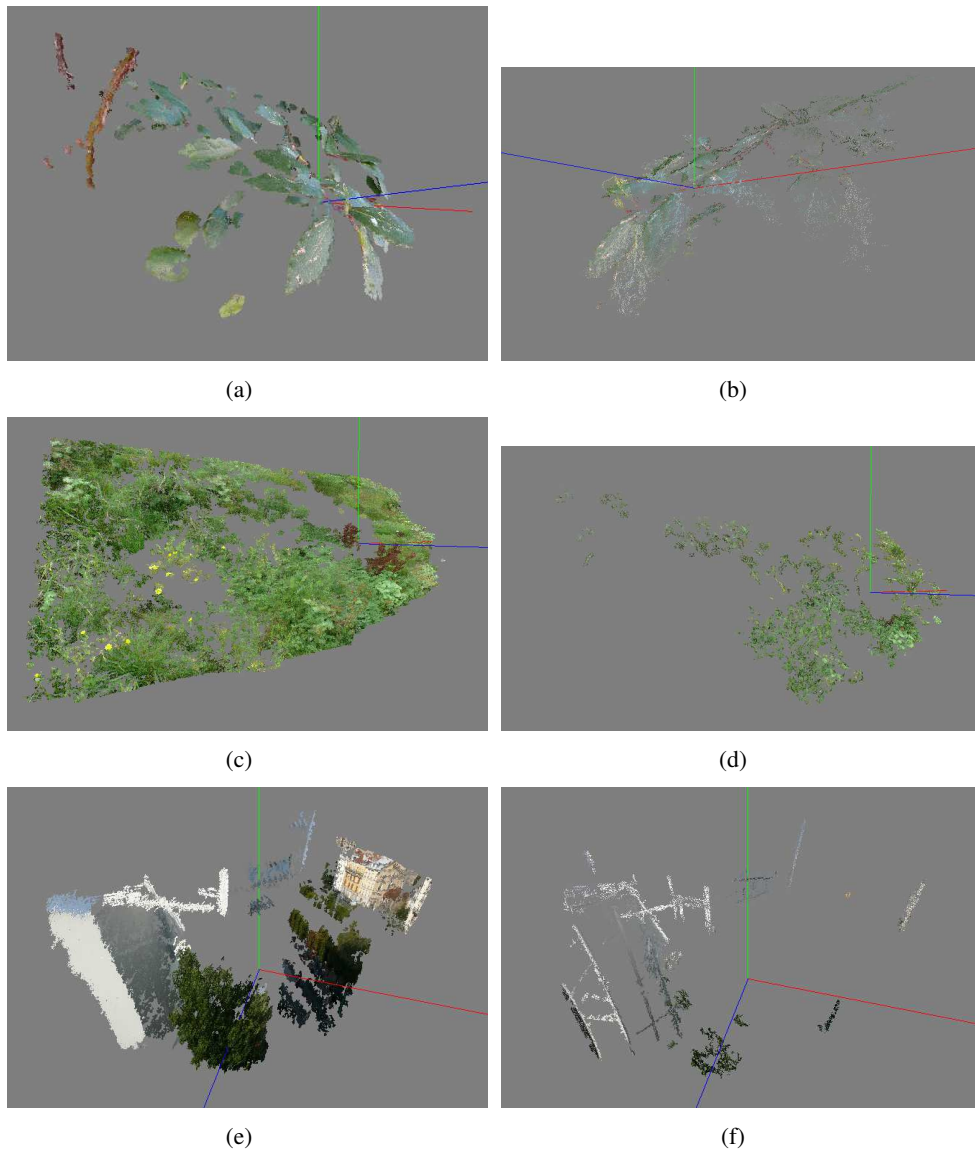


FIGURE 3.1.6 – Comparaison entre l’algorithme de Furukawa en (b),(d) et (f) et notre chaîne complète de traitement en (a),(c) et (e). Le manque de points sur les nuages reconstruits par l’algorithme de Furukawa est du à la piètre qualité du calibrage et au problème de reconstruction de végétaux pour lesquels trop de différences existent entre les photos.

d’un visualisateur sous OpenGL). Les mesures réelles, quant à elles, seront réalisées à l’aide d’un pied à coulisse. Nous développerons donc dans ce chapitre, le banc de test réalisé pour comparer les mesures réelles et les mesures faites sur le nuage 3D. Nous montrerons les résultats obtenus dans la partie résultat qui nous permettront de valider la méthode.

3.2.1 Banc de test

Pour réaliser un banc de test, il faut pouvoir faire aisément des mesures réelles et virtuelles. Pour cela plusieurs choix s'offraient à nous :

- Réaliser une scène virtuelle dont les paramètres de calibrage et d'éclairage sont parfaits. Faire différentes acquisitions de la scène, puis utiliser notre chaîne afin de vérifier la validation des mesures faites.
- Utiliser des photos d'une scène particulière pour obtenir la meilleure reconstruction possible. Le problème venant du choix de la scène à reconstruire.

Nous avons décidé de ne pas travailler sur des scènes virtuelles car, si les résultats devaient être validés seule une partie de notre chaîne serait alors validée. La robustesse aux problèmes liés aux photos ne serait pas vérifiée.

Choisir d'utiliser une scène réelle permettait de vérifier l'intégralité de la chaîne dans des conditions normales de prises de vues. Il nous restait à choisir la scène dont tous les paramètres sont contrôlés et sur laquelle nous pourrions faire des mesures. La première idée fut d'utiliser des objets de forme géométrique. Mais la reconstruction de tels objets implique la mise en lumière de certains types d'erreurs qui sont peu présentes dans les reconstructions de scènes naturelles. Il fut donc choisi d'utiliser des objets utiles à notre problématique comme des galets. Les galets utilisés proviennent de la plage de Pordic en Bretagne, qui ont été disposés sur une planche recouverte de papiers millimétrés pour permettre des mesures de vérification directement sur les photos, visible sur la figure 3.2.1.

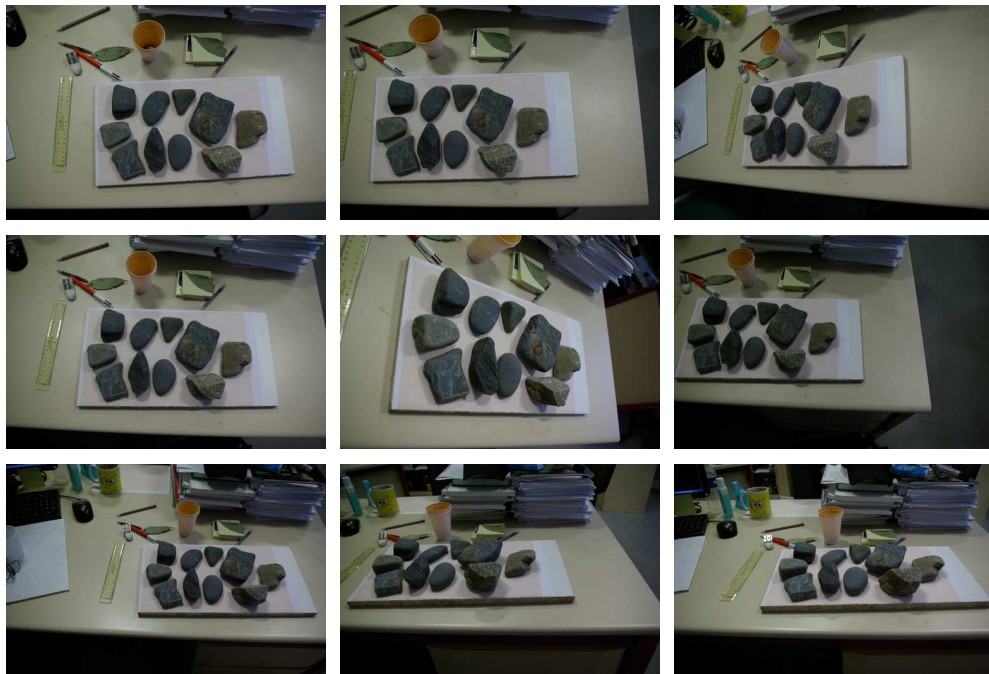


FIGURE 3.2.1 – Exemples de photos des galets prises sur un bureau avec un mauvais éclairage.

3.2.2 De l'espace métrique à l'espace euclidien

Faire des mesures sur les nuages 3D est toujours impossible. Le problème est que notre reconstruction n'est pas euclidienne mais métrique, c'est-à-dire que les distances mesurées sur le nuage ne correspondent pas à des mesures en centimètres mais à une grandeur sans dimension. Et même si ces mesures restent vraies, elles le sont à un facteur d'échelle près :

$$D_{\text{métrique}} = \lambda D_{\text{euclidienne(cm)}}$$

avec $D_{\text{métrique}}$ une distance mesurée dans un espace métrique, $D_{\text{euclidienne(cm)}}$ la même distance mais en centimètre et λ le facteur d'échelle entre les deux espaces.

Il suffit donc de connaître ce facteur pour pouvoir passer de l'espace métrique à l'espace euclidien. Or, une manière de calculer ce facteur est de connaître une correspondance entre une mesure de l'espace réel et une de l'espace 3D. Il suffit alors de faire le rapport entre les deux mesures pour trouver le facteur d'échelle λ . Sur le banc de test, c'est le papier millimétré qui servira à cette mesure. La mesure sur les surface 3D s'effectuera en choisissant manuellement les deux extrémités de la longueur à extraire.

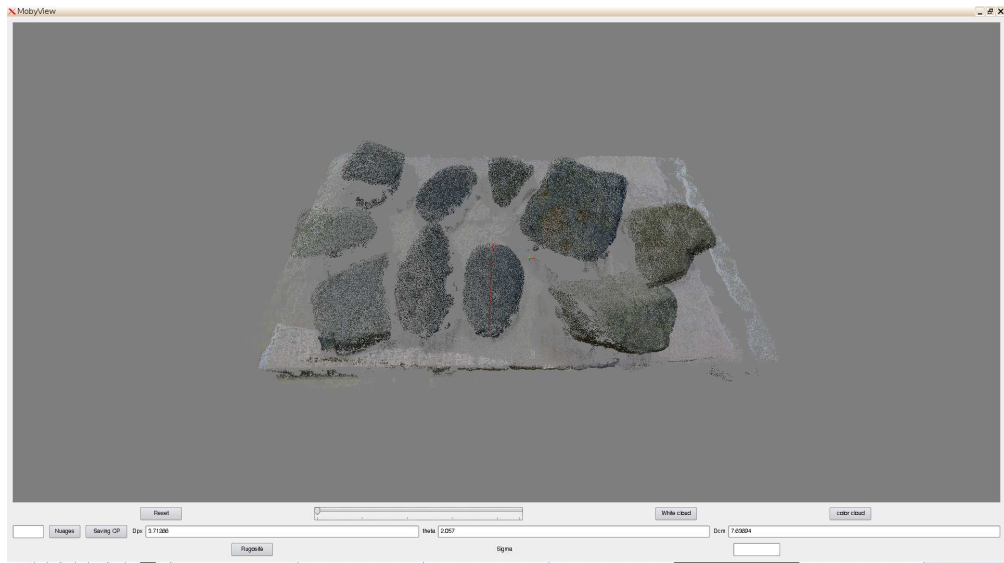
Une fois le facteur d'échelle trouvé, on peut effectuer des mesures qui seront automatiquement transformées en mesures centimétriques que l'on pourra comparer à des mesures faites au pied à coulisse comme montrée sur la figure 3.2.2.

3.2.3 Résultats des tests

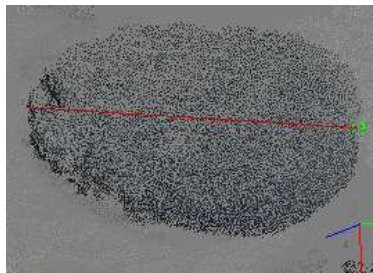
Des mesures ont été faites sur le banc de test dont les résultats sont résumés dans le tableau 3.1. Les mesures réelles étant faites au pied à coulisse sans affichage numérique, il est difficile de mesurer au-delà du millimètre. Finalement, il fut choisi de rajouter 0.05 cm à la mesure au pied à coulisse si le curseur est entre deux millimètres. Les mesures 3D seront arrondies au dixième de millimètres (au delà l'intérêt est limité).

Type de mesures 3D	Mesures 3D(cm)	Mesures réelles(cm)	erreur (mm)	erreur (%)
Longueurs	7.81	7.65	1.6	2.09
-	7.63	7.75	1.2	1.55
-	8.39	8.30	0.9	1.08
-	5.92	5.75	1.7	2.95
-	8.09	7.90	1.9	2.40
-	6.85	6.70	1.5	2.23
Épaisseurs	1.65	1.85	2.0	10.81
-	1.94	2.10	1.6	7.62
-	1.54	1.30	2.4	18.46
-	3.33	3.10	2.3	7.42
-	1.73	1.80	0.7	3.89
-	1.83	2.10	2.1	10

TABLE 3.1 – Exemples de mesures faites sur le nuage de cailloux 3.2(a).



(a)



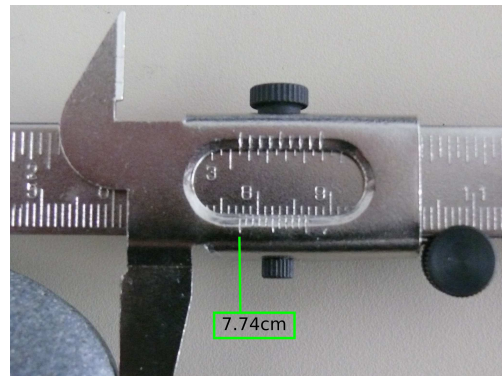
(b)



(c)



(d)



(e)

FIGURE 3.2.2 – Exemple de mesures faites sur le nuage et en réalité. (a) Mesures faites directement sur le nuage 3D, (b) gros plan de la fenêtre de mesures, (c) exemple de mesure faite au pied à coulisse, (d) gros plan sur la mesure, on s'aperçoit que l'erreur est d'un peu plus d'un millimètre (différence entre 7.63cm et 7.75 cm).

Les mesures sur le tableau 3.1 sont séparées pour faire ressortir les erreurs en fonction du type de mesures, dans le sens de la longueur ou de la profondeur (différence

entre la planche et le sommet du caillou).

Les erreurs semblent moins grandes sur les longueurs que sur la profondeur. En effet la moyenne des erreurs sur les longueurs est de 1.5 mm et pour l'épaisseur de 1.8 mm et le pourcentage d'erreur étant dans le cas des longueurs de 2.05% et pour l'épaisseur de 9.7%. Les erreurs sont donc globalement plus grandes sur les mesures de profondeur, même si cela reste faible.

Pour expliquer ces mauvaises mesures de profondeurs, nous avons fait deux hypothèses :

- La mesure faite sur le nuage 3D peut ne pas être parfaitement perpendiculaire au plan. Ce problème vient de l'incertitude du choix des points appartenant à la planche et du point du caillou censés être au dessus de ceux-ci, comme il est possible de le voir dans la figure 3.2.3.
- La précision n'est pas la même en planimétrie qu'en profondeur.

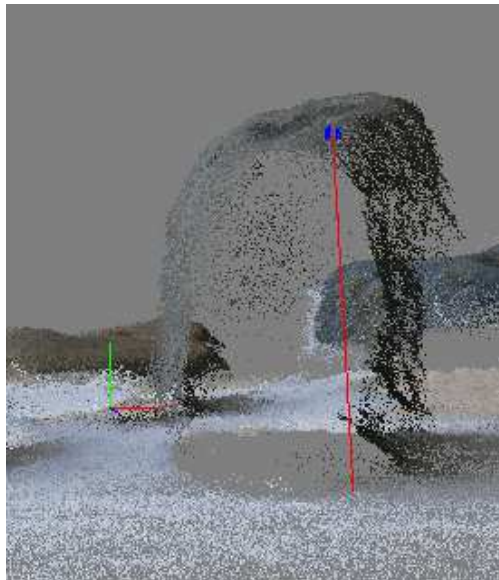


FIGURE 3.2.3 – Exemple d'une mesure sur la profondeur expliquant les moins bons résultats, car il est clair qu'il est difficile de déterminer un point du sol. La mesure est toujours un peu oblique et non orthogonale

Les deux mesures avec les plus grandes erreurs ont été trouvées sur les deux cailloux les plus éloignés de l'objectif de l'appareil photo. Ce phénomène est compréhensible, car, comme nous l'avons expliqué dans le chapitre sur la stéréo-vision, plus l'objet est loin par rapport à la ligne de base et moins la précision est bonne.

Au vu de ces résultats, nous avons choisi de refaire une étape de mesure sur un nouveau nuage, produit à partir des mêmes pierres, mais dont les photos sont prises avec un meilleur éclairage et avec un angle de vue plus fronto-parallèle à la planche, comme il est possible de le voir sur la figure 3.2.4.

Pour permettre une meilleure comparaison des résultats, nous avons choisi de garder les mêmes mesures réelles faites au pied à coulisse. Les longueurs sur les cailloux n'ont été faites qu'une fois, nous n'avons refait que les mesures sur le nuage



FIGURE 3.2.4 – Résultats obtenus pour des photos prises avec un meilleur éclairage.

3D.

Types de mesure 3D	Mesures 3D(cm)	Mesures réelles(cm)	erreurs(mm)	erreurs(%)
Longueurs	7.72	7.65	0.7	0.91
-	7.65	7.75	1.0	1.29
-	8.36	8.30	0.6	0.72
-	5.93	5.75	1.8	3.13
-	8.02	7.90	1.2	1.52
-	6.79	6.70	0.9	1.34
Épaisseurs	1.71	1.85	1.4	7.56
-	1.97	2.10	1.3	6.19
-	1.49	1.30	1.9	14.61
-	3.32	3.10	2.2	7.09
-	1.74	1.80	0.6	3.33
-	1.91	2.10	1.9	9.04

TABLE 3.2 – Mesures faites sur le nuage de cailloux de la figure 3.2.4.

Les mesures effectuées sur le nouveau nuage de point, vu en 3.2.4, montrent des résultats légèrement meilleurs. L'hypothèse que nous faisons est que cette amélioration est liée au nombre d'images, plus nombreuses que dans la première validation avec de meilleures conditions de prises de vues. Elles ont été prises avec un meilleur second plan permettant d'obtenir un maximum d'informations sur la base des cailloux. Mais la disparition du papier millimétré nous a obligé pour le calcul du λ à mesurer une longueur d'un caillou non utilisé pour la validation.

Nous avons voulu vérifier si les résultats des mesures étaient différents avec l'algorithme PMVS2 [Furukawa and Ponce, 2009]. Or, les résultats trouvés sont sensiblement équivalents, les mesures effectuées sur le nuage présenté sur la figure 3.2.5 ont donné une erreur moyenne de 1.6mm en longueur et 1.7mm en épaisseur.

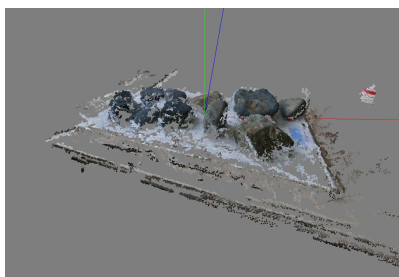


FIGURE 3.2.5 – Banc de test reconstruit par l’algorithme PMVS2.

3.3 Discussions

Nous avons vu que les résultats obtenus permettaient de faire des mesures proches de la réalité, mais que ces mesures conservaient des erreurs. Elles sont de provenances diverses et qui doivent être étudiées dans cette partie, en particulier les erreurs provenant des méthodes utilisées.

Plusieurs hypothèses ont été mises en lumière pour expliquer ces erreurs, en sachant que les conditions de prises de vues sont considérées comme parfaites :

- un problème de mesure. C’est-à-dire que la mesure faite sur le nuage ne correspond pas à celle faite sur l’objet mesuré. Cela peut paraître simple mais en réalité faire correspondre au maximum ces deux longueurs est très difficile. Ainsi, il est possible que l’erreur vienne de la méthodologie utilisée pour faire les mesures. Celle-ci consiste à faire les mesures sur le nuage 3D puis à les reproduire le plus fidèlement possible sur l’objet mesuré.
- un problème dans l’extraction du facteur d’échelle λ . Son calcul est primordial et une erreur dans cette mesure entraîne la présence d’une erreur dans toutes les mesures. Il faut donc la calculer le plus finement possible. Or si une telle erreur devait intervenir, celle-ci serait équivalente dans toutes les mesures. Elle deviendrait alors facilement identifiable et serait facilement corrigée.
- un problème provenant de la chaîne de traitements.

Dans cette partie nous n’étudierons que les erreurs provenant des techniques utilisées, et impliquant des biais sur les mesures. Dans ce cadre nous tenterons d’estimer la part prise par ces problèmes dans les erreurs de mesures. De plus nous ferons part de nos observations sur la façon de faire des prises de vue capable d’offrir la meilleure reconstruction possible.

3.3.1 Étude de l’erreur de disparité

Nous avons choisi de séparer le problème en deux parties :

- la provenance des erreurs de disparité
- l’influence de ces erreurs sur la position 3D.

Raisons ...

Les causes d'erreurs de disparité sont nombreuses et sont généralement éliminées par les étapes de filtrage du nuage 3D. Ces erreurs proviennent pour la plupart des hypothèses effectuées pour la mise en correspondance :

- Si le recouvrement entre les images n'est pas assez fort. En effet, un point doit être vu par au moins 2 images c'est-à-dire qu'il faut un recouvrement d'au moins 50% entre deux images successives. Pour une reconstruction optimum, l'expérience nous a montré qu'un recouvrement de 60 à 80% donnait un meilleur résultat.
- Si les photos sont floues, l'information se retrouve alors fortement bruitée. Sous-échantillonner les images est une solution mais la disparité restera fortement bruitée.
- Si la distance entre deux prises de vue est trop petite pour la distance à la scène prise en photo (un trop petit B/H). La disparité sera alors imprécise. Il est arrivé pour les photos prises avec dix centimètres de base pour une scène prise à 20 mètres de hauteur que la disparité n'ait que deux valeurs possibles.
- Prendre les photos avec trop d'effet perspectif, c'est-à-dire prendre une surface plane de manière rasante provoque énormément de mauvais appariements. Si ce problème est couplé avec des motifs répétitifs, on obtient un très mauvais résultat qui n'est pas identifié par nos algorithmes comme de mauvais appariements.
- La présence de surfaces uniformes provoque de nombreux problèmes de mise en correspondance. Ainsi pour la deuxième validation et pour permettre une bonne reconstruction de la planche sur laquelle repose les galets, celle-ci a été crayonnée de manière à permettre une bonne mise en correspondance.
- La présence de structures répétitives provoque de mauvaises correspondances liées aux trop nombreux pics de corrélation, comme par exemple avec le papier millimétré qui a produit quelques zones mal mises en correspondance.
- Une des hypothèses faite lors de l'appariement est que les surfaces reconstruites sont lambertiennes, c'est-à-dire que les surfaces diffusent la lumière de manière homogène dans toutes les directions. Ces surfaces n'existent pas, mais des surfaces que l'on appellera "mates" répondent bien à la corrélation, contrairement aux surfaces spéculaires ou "brillantes" qui sont elles très difficiles à mettre en correspondance.

Les problèmes peuvent venir de la scène elle-même. Si une scène présente un trop grand nombre de plans se chevauchant, alors trop d'occlusions seront présentes. Or, la présence de ces plans entraîne ordinairement des phénomènes d'adhérence. Ce qui peut être problématique, même si normalement ils sont supprimés par l'utilisation d'images de segmentation. Mais dans certains cas, les adhérences se sont maintenues. De même un mauvais choix des paramètres de l'algorithme de mise en correspondance dense provoquera souvent un manque de corrélations et des erreurs de disparité. Ainsi, même si l'on remarque peu d'erreurs de ce type grâce notamment à la robustesse des algorithmes choisis et aux filtrages effectués, il est important d'étudier les conséquences de telles erreurs. Nous ne tenterons pas de mesurer la quantité d'erreurs de correspondance, sachant que ces erreurs sont très variables en fonction de la scène observée, nous nous contenterons d'en examiner les conséquences sur les mesures.

... et influences

Pour étudier les conséquences d'erreurs de disparité, il suffit d'introduire des erreurs de corrélations et de regarder leurs influences sur la position du point 3D. Pour cela il faut connaître des points pour lesquels une grande confiance existe dans leur appariement. Il suffit alors de les projeter, puis d'introduire une erreur d'appariement et enfin de les reprojeter de la même manière. On calcule alors la distance entre les deux points, cette distance sera considérée alors comme l'erreur liée à la corrélation. Nous ferons varier l'erreur pour observer l'évolution de l'erreur 3D. Nous étudierons de plus l'influence de l'erreur de mise en correspondance dans des mesures connues.

Pour les points de confiance, les appariements obtenus par SURF semblent être un bon choix. C'est-à-dire que la comparaison se fera entre les projections d'un appariement erroné et d'un appariement dans lequel nous avons confiance. Une moyenne des erreurs 3D sera calculée pour toutes les erreurs introduites sur les appariements SURF. Cette moyenne notée $\bar{\varepsilon}$ sera calculée sur l'erreur de distance 3D, ε .

$$\bar{\varepsilon} = \frac{1}{N} \sum_{i=1}^N d(X_k(i), X_{origine}(i))$$

Avec k l'erreur en pixel introduite dans la disparité, N le nombre de points SURF trouvés et considérés comme inliers, d la distance euclidienne, X_k le point 3D trouvé pour le couple erroné, $X_{origine}$ le point 3D projeté à partir du couple SURF.

On voit bien sur les figures que l'erreur de mise en correspondance a peu de conséquence sur les mesures. Même si les erreurs doivent être minimisées, il semble clair que ce n'est pas le calcul de la disparité qui provoque les erreurs de mesures observées.

3.3.2 Étude de l'erreur de calibrage

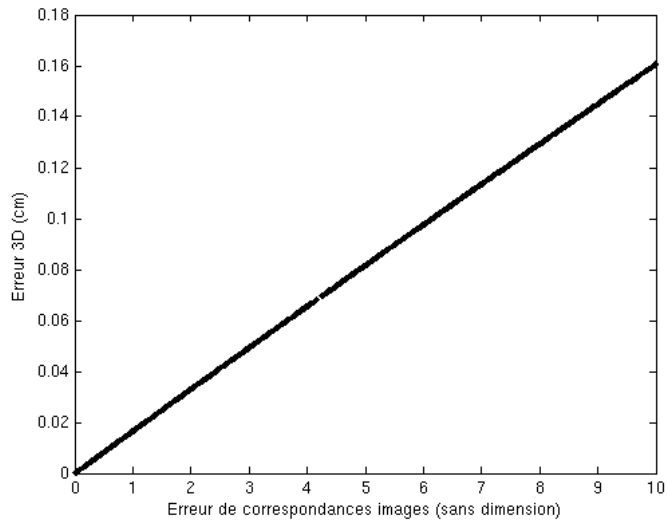
Si les erreurs ne viennent pas des erreurs de mise en correspondance, une autre hypothèse est que ces erreurs viennent de l'auto-calibrage possédant de nombreuses faiblesses par rapport à un calibrage sur mire.

Nous étudierons donc les erreurs possibles de calibrage. Or, nous avons vu que pour le calcul des paramètres intrinsèques et extrinsèques, un certain nombre d'hypothèses ont été faites, notamment sur les paramètres intrinsèques. Ces hypothèses sont :

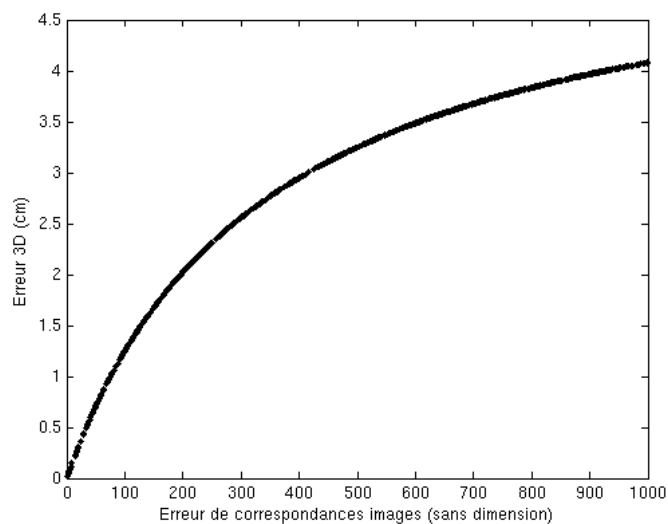
- la forme carrée des pixels ;
- la position du point d'auto-colimation au centre de l'image ;
- la focale venant des EXIF, dont les constructeurs nous annoncent un certain pourcentage d'erreur ;

Les paramètres extrinsèques peuvent aussi être mal estimés. Ces phénomènes provoquent une mauvaise position et orientation de la caméra. Donc, lors de la triangulation, les points seront mal placés dans l'espace 3D.

Nous tenterons ici de voir les conséquences sur les mesures de ces erreurs de calibrage. Dans un premier temps nous étudierons le comportement d'une erreur sur la distance focale, puis l'influence des erreurs d'estimation des paramètres extrinsèques : du vecteur de translation et de la matrice de rotation.



(a)



(b)

FIGURE 3.3.1 – Schéma des rapports trouvés entre les erreurs de mise en correspondance sur les images et les erreurs résultantes sur le nuage 3D. (a) Pour une erreur inférieure à 10 pixels, on s’aperçoit que la valeur de l’erreur en cm est négligeable, en sachant qu’une erreur de 10 pixels reste un cas rare, (c) dans le cas extrême jusqu’à 1000 pixels d’erreurs (outliers) on s’aperçoit du comportement logarithmique de l’erreur, ceci lié à l’imprécision de la profondeur à partir d’une certaine disparité.

Influence de la focale

Pour étudier l’influence d’une erreur sur la distance focale, il suffit d’introduire une erreur et d’observer l’évolution des positions des points 3D. Pour introduire

cette erreur, nous tronquerons et augmenterons la valeur de focale. Pour maintenir une certaine cohérence dans les résultats, les mêmes points utilisés pour le calcul de l'erreur de disparité seront choisis. La même méthodologie servira à calculer la moyenne des erreurs.

Lorsque la longueur focale est augmentée (voir figure 3.3.2), on s'aperçoit que l'erreur de position à un comportement logarithmique jusqu'à une certaine valeur asymptotique. Cette dernière correspond au cas d'une focale de longueur infinie.

On observe sur la figure 3.3.3 que les mauvais résultats, obtenus pour une longueur focale inférieure à $\frac{f}{5}$, correspondent au moment où la focale devient trop petite et où les rayons ne se croisent plus.

Nous ne pouvons pas réellement tester la vérité terrain d'une focale. Les constructeurs fournissent un ordre de grandeur de l'erreur de 2% de la longueur focale (fournit par Lumix et Pentax). Cette erreur étant donc souvent très petite, l'erreur de la position 3D est généralement inférieures au demi-centimètre (voir figure 3.3.4).

Ces résultats nous ont motivés dans le choix de ne pas ré-estimer la longueur focale et de garder celle transmise dans les EXIF.

Influence de la translation

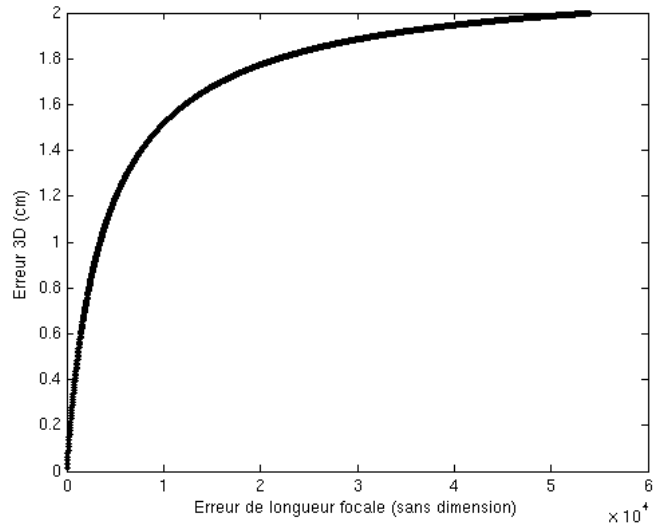
Sans connaître une vérité terrain de la géométrie de prise de vue, nous ne pouvons estimer l'erreur mais simplement en observer les conséquences. Pour cela nous avons choisi de séparer le problème en différentes parties :

- introduire une erreur sur un seul axe à la fois, ce qui a pour conséquence de changer la direction de prise de vue (voir figure 3.3.5(a)) ;
- introduire une erreur de translation dans la direction de la translation, c'est-à-dire que l'image est reculée ou avancée par rapport à sa position initiale. Cette erreur ne change pas la direction de prise de vue (voir figure 3.3.5(b)) ;

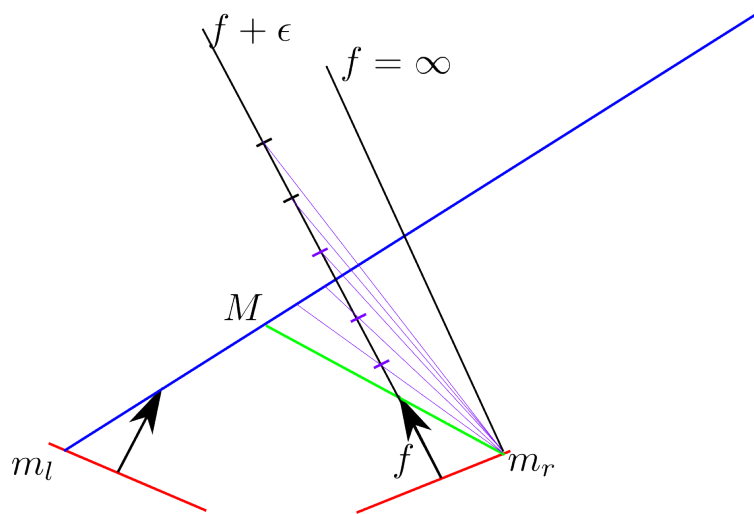
Puisqu'il est impossible d'avoir une vérité terrain, nous avons choisi d'exagérer l'erreur pour observer le comportement du nuage dans un tel cas. Une telle erreur n'a, empiriquement, aucune réalité et dans les rares cas où de elles se sont produites lors de l'auto-calibrage, les nuages étaient très facilement identifiables et donc supprimables.

On remarque sur la figure 3.3.5 que l'erreur de translation est une composante importante de l'erreur finale. En effet, même une erreur faible entraîne des erreurs qui peuvent atteindre un centimètre, ce qui est important comparé aux résultats obtenus précédemment. Dans notre validation nous avons obtenu des erreurs de mesure de l'ordre du millimètre, ce qui nous laisse penser que les erreurs réelles de la translation sont inférieures.

Sur la figure 3.3.5(a) l'erreur principale est donnée par l'axe X, ce qui est consistant avec les prises de vues de droite à gauche (l'axe X). Entre deux images, il y a peu de modifications en Y ou Z mais une très grande en X. Ainsi introduire une erreur dans la direction de la translation entre les photos influe beaucoup sur la position finale des points 3D.



(a)

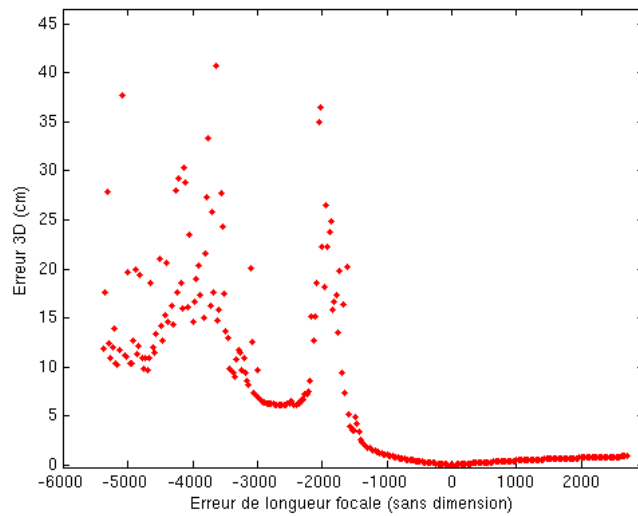


(b)

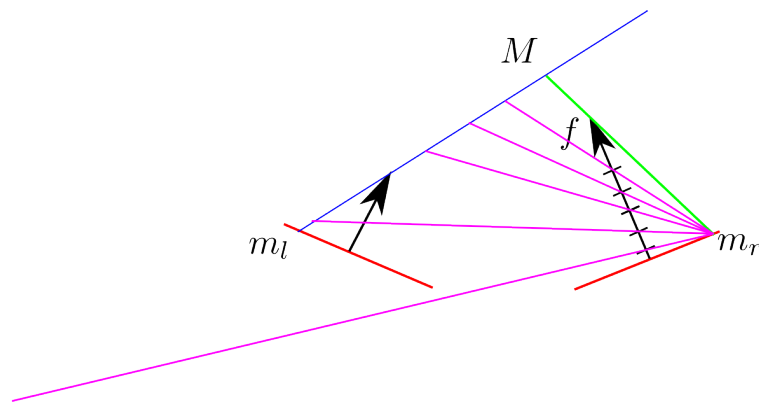
FIGURE 3.3.2 – (a) Schéma de l'influence d'une augmentation de longueur focale sur la position 3D. L'erreur ne dépasse pas 2cm, ce qui est beaucoup en comparaison des erreurs de corrélations. Ces valeurs d'erreur sont obtenues pour un ajout de plus de 400% de la longueur focale. (b) Schéma expliquant le comportement logarithmique de l'erreur focale. Les lignes violettes représentent la projection dans l'espace du point m_r . Quand la longueur focale devient très grande, le point d'intersection des lignes violettes avec la ligne bleue, représentant la triangulation de m_l , devient de plus en plus proche de la ligne noire. Cette asymptote correspond à une focale à l'infini.

Influence de la rotation

Pour les mêmes raisons que précédemment nous nous contenterons d'introduire une erreur et d'en observer les conséquences. Pour introduire une erreur, nous avons



(a)



(b)

FIGURE 3.3.3 – (a) Schéma de l’influence de l’erreur de longueur focale sur la position 3D, ici la longueur focale va de $\frac{1}{2}f$ à $2f$, avec f la longueur focale. On observe un comportement très mauvais lorsque l’on tronque la longueur focale d’une valeur trop grande. (b) Schéma expliquant le problème lié à la diminution de la focale, quand les deux demi-droites ne se coupent plus. La triangulation du point a alors un comportement erratique et l’erreur devient très grande.

fait le choix d’ajouter une rotation d’un certain angle selon les trois axes du repère 3D et selon la direction de prise de vue. Pour introduire cette rotation, nous multiplierons la matrice originale par une matrice de rotation. Le but est de changer l’orientation de prise de vue comme il est possible de le voir dans la figure 3.3.6.

Ne connaissant pas les erreurs existantes dans les reconstructions, nous avons décidé d’exagérer les erreurs de rotations.

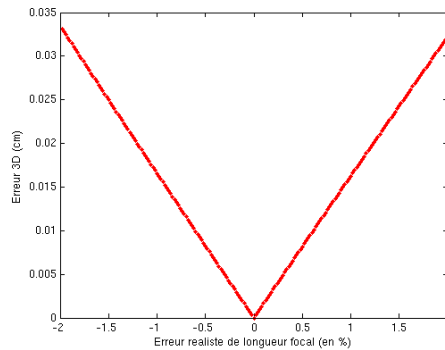


FIGURE 3.3.4 – Rapport de l’erreur de la focale sur la position 3D en prenant compte des informations constructeurs, on remarque que les erreurs ne dépassent pas les 5 millimètres.

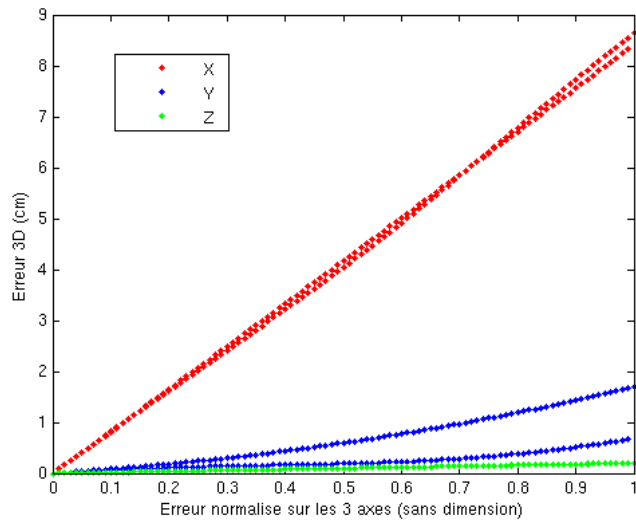
On aperçoit sur les courbes de la figure 3.3.8, que modifier la rotation autour des 3 axes entraîne de très fortes erreurs allant jusqu’à 20 centimètres. Le long de l’axe de prise de vue, les modifications provoquent des erreurs allant jusqu’à 7 centimètres, ce qui est moins important. On observe ainsi des erreurs importantes et bien supérieures à celles observées sur la validation, une erreur de 5 degrés entraînant un biais de mesure d’un centimètre. On peut ainsi en déduire que les erreurs d’estimation de la rotation sont faibles.

Erreurs de calibrage

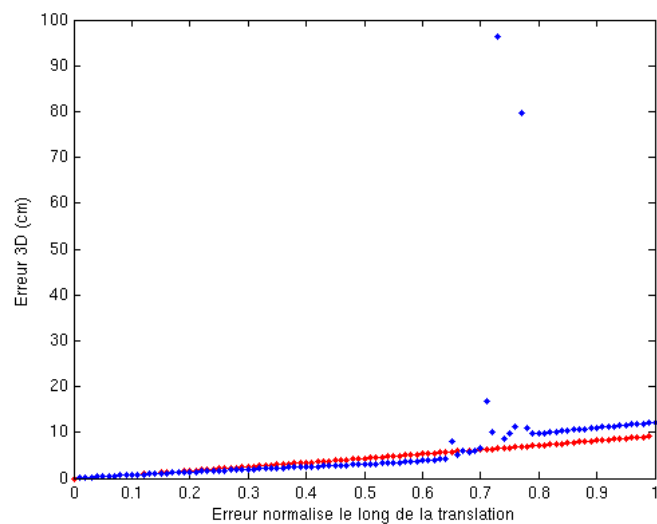
Nous avons vu que les erreurs provenant du calibrage influencent beaucoup la position des points 3D, spécialement dans le cas d’erreurs des paramètres extrinsèques. Nous nous sommes aperçu que les erreurs de calibrage sont, soit très fortes, et sont alors identifiables, soit très faibles. Mais leur implication quoi qu’il arrive est grande. L’hypothèse que nous faisons est que les trois erreurs montrées dans ce chapitre se cumulent. C’est-à-dire qu’il existe de très faibles erreurs de focale, de translation et de rotation, qui cumulées entraînent des erreurs de mesures observées dans la validation. Ainsi lors de notre validation, nous avons fait attention de supprimer les images qui semblaient ne pas bien être calibrées. Ce type de vérification est parfois fastidieuse, mais permet d’avoir une meilleure reconstruction. Nous avons vu qu’une erreur sur le calibrage entraîne de très fortes modifications du nuage 3D. Ainsi, de la réussite de l’étape de calibrage dépend la qualité du nuage final. Or, le calibrage est de bonne qualité si les images sont prises en respectant certaines conditions.

3.3.3 Conseil de prises de vues

Au vu des problèmes exposés, nous tenterons d’utiliser nos connaissances théoriques et empiriques sur la provenance des erreurs pour donner des conseils aux personnes susceptibles d’utiliser ces méthodes dans le but de permettre la meilleure reconstruction possible.



(a)



(b)

FIGURE 3.3.5 – (a) Schéma de l’influence de l’erreur de translation, dans les trois directions X, Y, Z , dans la position 3D. Les courbes sont doubles car les valeurs sont ajoutées et soustraites, (b) Rapport de l’influence d’une erreur de translation dans la direction de la translation, c’est-à-dire en diminuant et augmentant la norme du vecteur. Ainsi, en diminuant la norme de la translation (en rouge) puis en l’augmentant (en bleue), nous obtenons des résultats sensiblement équivalents, sauf au delà d’une certaine augmentation définie sur la plupart des cas à la moitié de la longueur de translation, où le comportement des erreurs varie beaucoup et provoque des erreurs très importantes (jusqu’à un mètre)

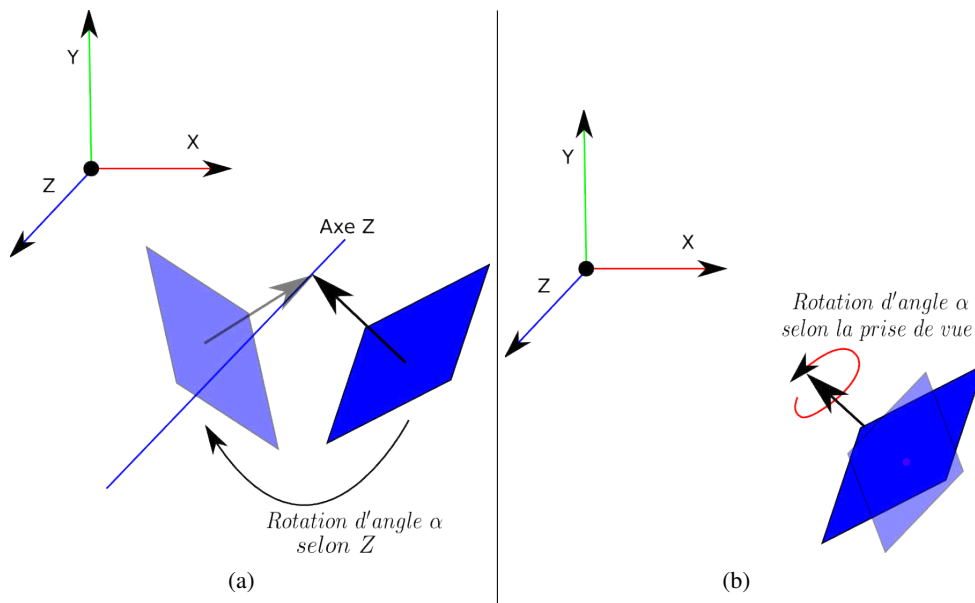


FIGURE 3.3.6 – (a) Schéma de la rotation selon les axes, avec ici une rotation autour de l'axe Z, (b) schéma de la rotation selon l'axe de prise de vue

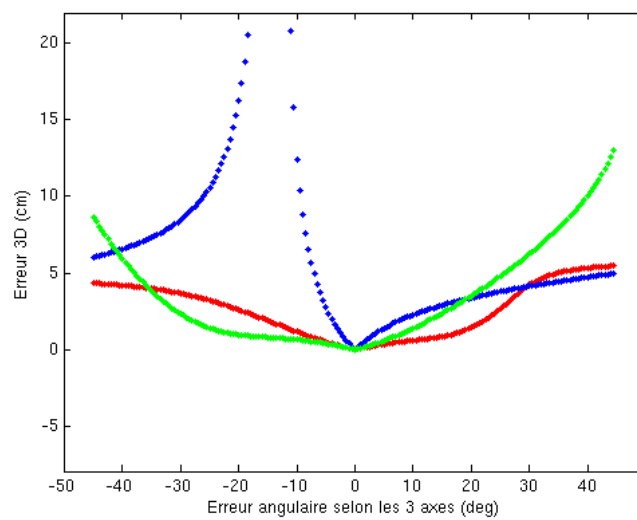


FIGURE 3.3.7 – Rapport de l'erreur de rotation des 3 axes originaux sur la position 3D, X en rouge, Y en bleu, Z en vert

Cette section a pour but de réunir les conseils permettant d'obtenir les meilleurs reconstructions possibles.

- Utiliser une taille d'image adaptée au capteur, il est inutile d'avoir de trop grandes images si la qualité n'est pas intéressante. Il faut toujours privilégier la qualité à la quantité.

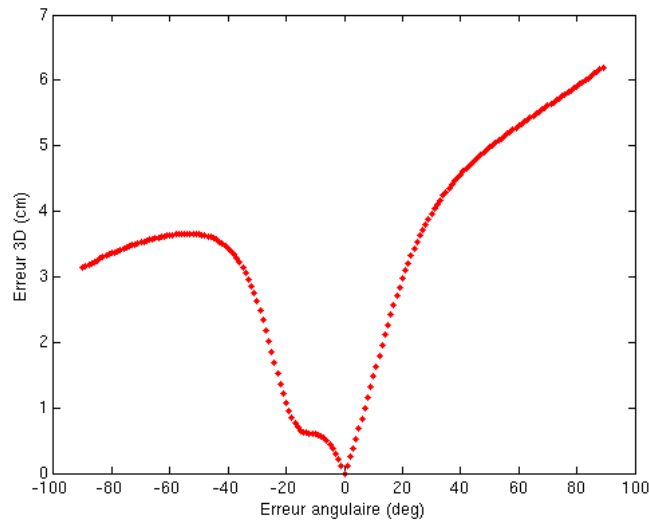


FIGURE 3.3.8 – Rapport de l’erreur de rotation autour de l’axe de prise de vue sur la position 3D

- L’illumination se doit d’être importante pour éviter la présence de trop de bruit dans les images. Le cas idéal suppose une illumination diffuse pour éviter la présence d’ombres.
- Éviter le flou dans les images, pour cela le temps d’exposition ne doit pas être trop important, ce qui rejoint le point précédent. Dans le cas d’une illumination moins forte, utiliser de préférence un support pour l’appareil.
- Avoir un recouvrement d’au moins 75% entre les images pour un résultat optimum.
- Éviter de tourner l’appareil entre deux prises de vues. Privilégier les translations latérales.
- Éviter les sujets mouvants et dans le cas de feuilles tenter de prendre les photos par temps sans vent.

Conclusion

Les mesures effectuées nous ont confortées dans l’idée que la chaîne choisie offrait une véritable généricité et une certaine robustesse. Les mesures sur les cailloux seront complétées par des mesures faites sur des feuilles dont les photos ont été prises en condition extérieure. S’il est dommage de n’avoir pu tester la chaîne par rapport à une vérité terrain qui nous aurait permis de vérifier le nuage indépendamment de l’étape de mesure et ainsi de déterminer l’exactitude de la reconstruction, cela donne un ordre de grandeur des erreurs que l’on pourrait trouver en condition réelle avec des mesures peu précises et des erreurs inhérentes aux conditions de prises de vues.

En réalité, les conditions particulières de reconstruction de végétaux ont conditionné les différents choix algorithmiques. En effet, elles nous ont imposées de ne

choisir que les algorithmes les plus robustes, nous permettant d'être particulièrement générique vis-à-vis des scènes reconstruites. Cela permet au logiciel développé aux cours de la thèse d'avoir une grande portée au delà des applications directes visées par la thèse.

Enfin l'étape de discussion nous a permis de mettre en exergue le rôle prépondérant du calibrage dans les résultats obtenus. Elle nous a imposée de chercher la méthode la plus robuste et répondant le mieux aux contraintes. Il est apparu que l'algorithme d'auto-calibrage choisi, soutenu par le logiciel "bundler", prenait une place de plus en plus importante dans la communauté et s'imposait comme une référence. Une recherche bibliographique nous a permis de trouver quelques variantes de l'algorithme permettant une plus grande robustesse mais imposant un certain nombre de manipulations à faire par l'opérateur. Nous avons donc fait le choix de garder cet algorithme tel quel tout en connaissant la sensibilité des mesures aux erreurs de calibrage.

Maintenant que nous connaissons la qualité relative de nos nuages, il convient d'étudier les différentes mesures possibles pour l'aide aux photos-interprètes et aux chercheurs dont l'extraction de mesures pourrait être simplifiée par l'utilisation de notre méthode.

Pour cela nous mettrons en lumière deux domaines où l'extraction n'est pas aisée et pour lesquels l'utilisation de notre méthode pourrait apporter un plus non négligeable.

Chapitre 4

Mesures de paramètres de surfaces

Une fois connue la qualité des nuages reconstruits, des paramètres surfaciques peuvent en être extraits. La première question à laquelle nous avons du répondre fut celle des paramètres extractibles sur les points 3D. Or, le nuage 3D ne fournit pas seulement une information de profondeur, manquante aux photos, mais aussi une cohérence spatiale qui peut être utile. Le but de la thèse étant d'extraire des paramètres plus complexes que la valeur de profondeur mesurable directement sur le nuage, il nous fallait trouver d'autres paramètres à extraire.

Pour connaître les paramètres utiles nous avons collaboré avec des personnes impliquées dans l'utilisation de paramètres de surface. Deux problématiques principales, ayant une grande importance dans de nombreux phénomènes physiques, se sont dégagées de nos rencontres :

- la rugosité, dont l'importance est capitale dans de nombreux domaines allant de la rétro-diffusion radar à l'écoulement des eaux en terrain agricole.
- la dendrométrie, qui est un ensemble de paramètres complexes utiles pour l'étude de la bio-diversité.

Or, aujourd'hui, l'obtention de ces paramètres est compliquée et nécessite l'intervention de nombreux appareils et personnes. Nous proposons une nouvelle méthode rapide d'extraction de ces paramètres. Dans ce chapitre, nous montrerons comment nous avons calculé efficacement les différents paramètres de rugosité et de dendrométrie.

Pour chaque domaine nous montrerons leur utilité, leurs définitions, leurs manières "classiques" d'être extraites ainsi que les résultats obtenus.

4.1 Paramètres de rugosité

Il est intéressant de remarquer que la rugosité est un point clef dans l'étude de nombreux phénomènes physiques :

- la rétro-diffusion radar [Ulaby et al., 1979], [Zribi et al., 2000] ;
- l'érosion des sols [van Donk and Skidmore, 2003], [Zobeck and Onstad, 1987] ;
- l'hydrologie [Baghdadi et al., 2008], [Verhoest et al., 2008], [Engman and Chauhan, 1995] ;
- l'agriculture [Davidson et al., 2000], [van Donk and Skidmore, 2003],

[Baghdadi et al., 2008] ;

Pour ces différents phénomènes naturels, différents niveaux de détails sont nécessaires. Ainsi, pour définir la rugosité, il faut tout d'abord connaître le niveau d'échelle souhaité. Car en fonction de celui-ci, différents moyens existent pour extraire les paramètres de rugosité :

- à l'échelle microscopique (du millimètre au centimètre), la rugosité se calcule par l'étude *in-situ* des surfaces, que ce soit au moyen d'un rugosimètre à aiguilles ou laser (figure 4.1.1) ;
- à l'échelle mésoscopique (du décimètre au mètre), la rugosité s'obtient notamment par l'utilisation d'un LIDAR aéroporté ;
- à l'échelle macroscopique (du mètre au kilomètre), la rugosité s'obtient par l'utilisation d'un modèle numérique de terrain obtenu par imagerie aérienne [Smith et al., 2004] ;

Or, le principal problème des paramètres de rugosité est qu'ils sont difficiles à extraire quelque soit l'échelle. En effet, dans le cas macroscopique et mésoscopique, les moyens financiers demandés sont très importants. L'étude des paramètres de rugosité à cette gamme d'échelle nécessitant des systèmes aéroportés, ils ne seront pas étudiés dans cette section.

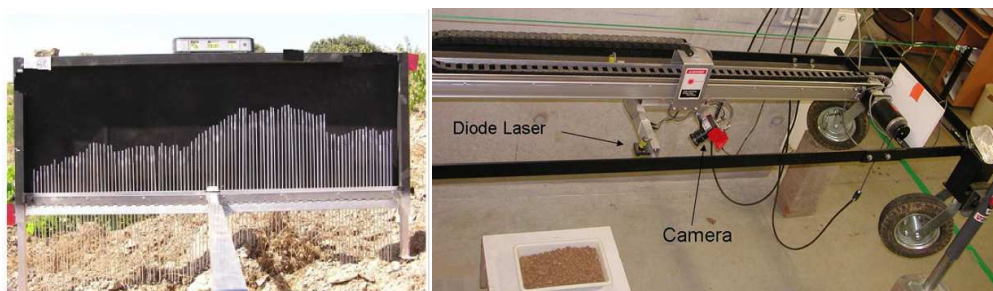


FIGURE 4.1.1 – Rugosimètres à aiguilles et laser.

Dans le cas microscopique, les appareils de mesure, vus dans la figure 4.1.1, posent un certain nombre de problèmes :

- ils ne peuvent prendre que des profils et d'une longueur limitée ;
- le rugosimètre à aiguille est très peu précis ;
- le rugosimètre laser est cher et difficile à transporter ;
- extraire les paramètres est long et fastidieux ;
- elles sont invasives pour la surface ;
- elles sont longues et difficiles à mettre en place ;

Le problème majeur de ces techniques est la prise de profils qui est la source de nombreuses erreurs comme expliqué dans l'article [Bryant et al., 2007].

Nous apporterons donc une contribution majeure dans l'étude de ces phénomènes physiques en proposant une nouvelle méthodologie d'extraction de paramètres de

rugosité en utilisant les nuages 3D [Petitpas et al., 2010].

Dans cette partie nous parlerons dans un premier temps des paramètres calculés. Puis, nous montrerons les utilisations possibles de la rugosité pour comprendre l'impact de mauvaises mesures et l'intérêt de notre apport. Enfin, nous étudierons ensuite les résultats obtenus.

4.1.1 Calcul de la rugosité

Nous nous sommes basés, pour les paramètres à calculer, sur les valeurs classiques de rugosité qui sont utiles à l'explication des phénomènes physiques à l'échelle microscopique [Verhoest et al., 2008].

Ainsi, l'état de l'art nous apprend que la définition la plus généralement acceptée est que la rugosité peut être obtenue par un ensemble de paramètres définissant la rugosité horizontale et verticale. Les paramètres expliquant la rugosité verticale sont l'écart type des hauteurs et parfois la moyenne arithmétique des hauteurs. Ceux qui déterminent la rugosité horizontale sont la longueur d'auto-corrélation et la fonction d'auto-corrélation. Ces paramètres sont définis par les équations (4.2), (4.3), et (4.4).

Toutes les mesures sont effectuées pour une rugosité de "moment 1", c'est-à-dire calculées par rapport à un plan moyen Π . Ce plan est choisi comme celui minimisant au sens des moindres carrés la distance euclidienne des points 3D à ce plan comme montré dans l'équation 4.1. D'autres moments peuvent être envisagés pour étudier la rugosité de plus haute fréquence mais n'ont pas été développés. Ainsi, sur notre nuage, il faut tout d'abord calculer l'équation du plan moyen :

$$\Pi = \arg \min_{\pi} \sum_{i=1}^n d(p_i, \pi)^2 \quad (4.1)$$

avec p_i est le i ème point 3D, d la distance euclidienne et π un plan quelconque.

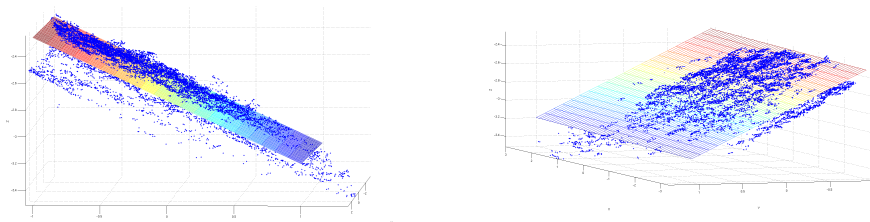


FIGURE 4.1.2 – Nuage de points 3D de la plage de galets et le plan moyen trouvé.

Enfin à partir de ce plan il suffit de calculer les distances des points 3D au plan moyen et de calculer les paramètres de rugosité suivant :

Sa : Moyenne arithmétique des hauteurs

$$S_a = \frac{1}{n} \sum_{i=1}^n |d(p_i, \Pi)| \quad (4.2)$$

Sq : Écart type des hauteurs

$$S_q = \sqrt{\frac{1}{n} \sum_{i=1}^n d(p_i, \Pi)^2 - \left(\frac{1}{n} \sum_{i=1}^n d(p_i, \Pi) \right)^2} \quad (4.3)$$

Où p_i est le i ème point 3D, Π le plan moyen et d la distance euclidienne.

Pour simplifier le calcul de la fonction d'auto-corrélation une image de profondeur est créée sur le nuage comme il est possible de le voir sur la figure 4.1.3.

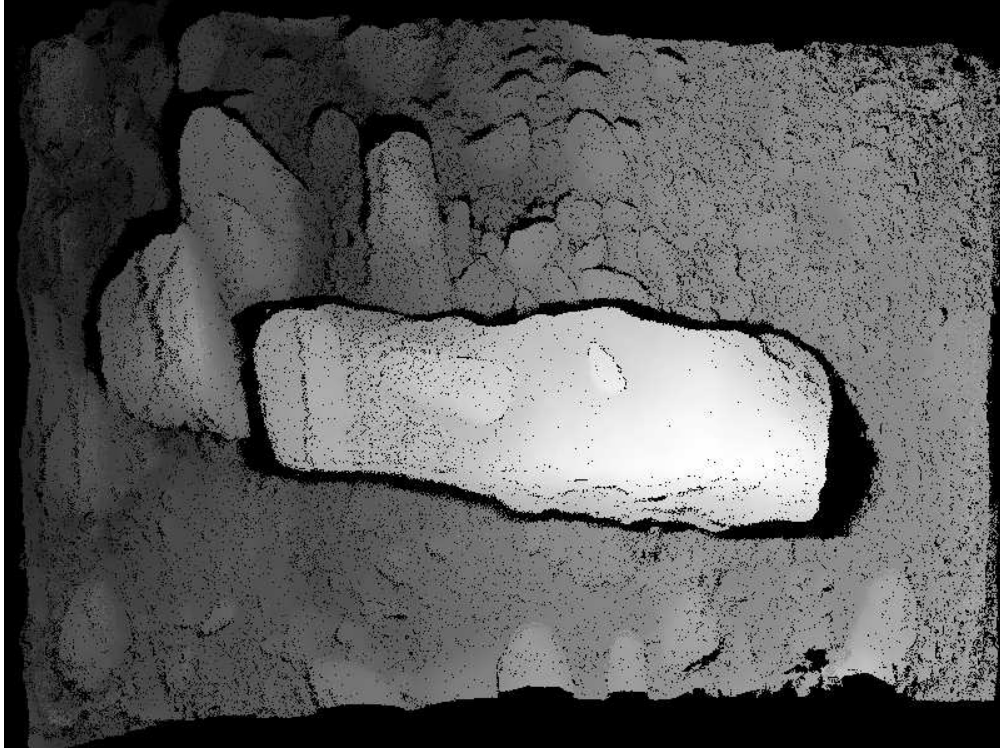


FIGURE 4.1.3 – Image de profondeur par rapport au plan moyen.

La fonction d'auto-corrélation (ACF) est calculée selon les deux directions horizontales. L'ACF centrée normalisée d'un signal discret $f_{i,j} = d(p_{i,j}, \Pi)$, où Π est le plan moyen, est définie comme :

$$ACF(m, n) = \frac{\sum_{i=1}^{M-m} \sum_{j=1}^{N-n} (f_{i,j} - \bar{f}) \cdot (f_{i+m,j+n} - \bar{f})}{\sum_{i=1}^{M-m} \sum_{j=1}^{N-n} (f_{i,j} - \bar{f})^2} \quad (4.4)$$

où \bar{f} est la valeur moyenne de f_i et N est le nombre de points sur l'axe X (soit le nombre de colonnes de l'image de profondeur), M le nombre de points sur l'axe Y (soit le nombre de lignes de l'image de profondeur).

A partir de cette fonction, la longueur d'auto-corrélation l , qui est importante dans l'explication de la rugosité, peut alors être exprimée sur les deux axes telle que :

$$ACF(l_x, 0) = \frac{1}{e} \quad ACF(0, l_y) = \frac{1}{e} \quad (4.5)$$

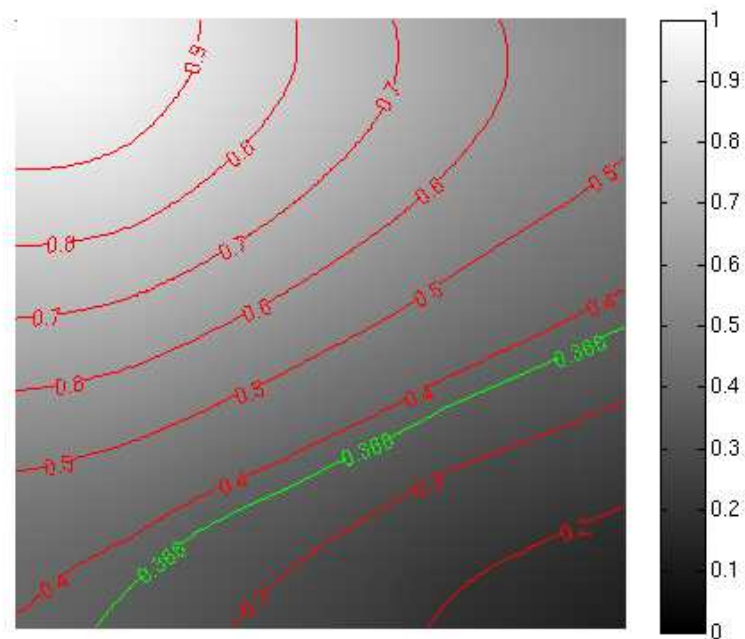


FIGURE 4.1.4 – Carte de la fonction d’auto-corrélation le long des deux axes x et y de l’image de profondeur, et en vert la courbe $1/e$, ce qui nous permet de calculer la longueur d’auto-corrélation sur les deux axes.

Il est possible de calculer $l(x,y)$ correspondant à la courbe de niveau de la fonction d’auto-corrélation à $\frac{1}{e}$ visible sur les figure 4.1.4 et 4.1.5(b).

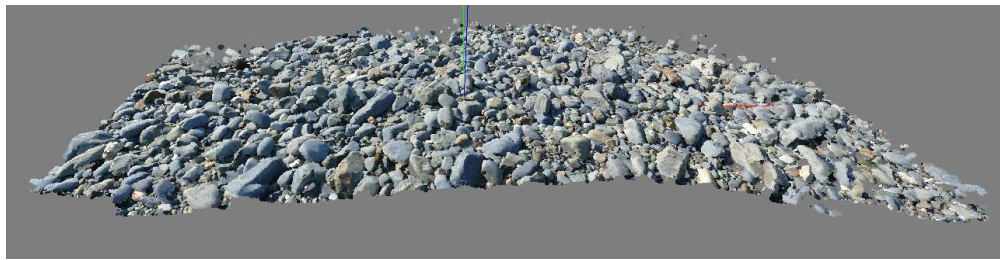
Sur la figure 4.1.5(b), la fonction d’auto-corrélation du nuage 3D, obtenue avec les images de galets, montre une anisotropie de la structure 3D, avec une longueur de 15.5 le long de l’axe x et 10.5 le long de l’axe y . L’explication possible de cette anisotropie est la forme des cailloux.

4.1.2 Utilité de la rugosité

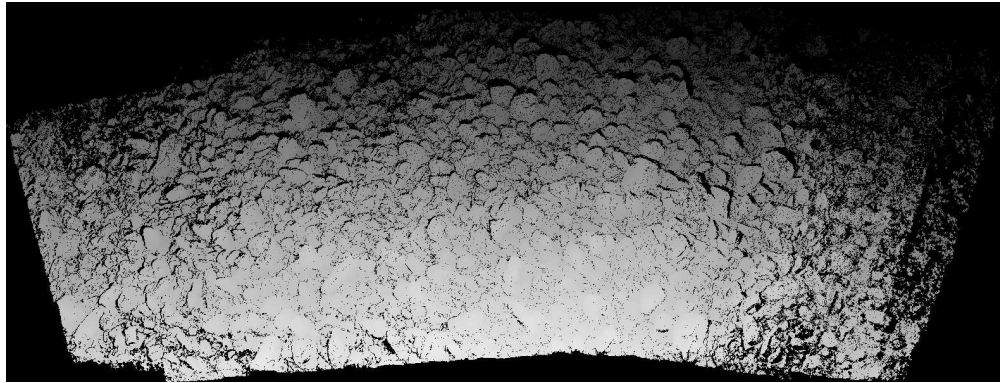
Nous étudierons ici quelques exemples connus de l’utilisation des paramètres de rugosité.

Les paramètres de rugosité interviennent pour expliquer de nombreux phénomènes naturels comme par exemple les avalanches [Margreth and Funk, 1999], les glissements de terrains [Dorren and Heuvelink, 2004], les inondations [Govers et al., 2000] ou le suivi forestier [Aubrecht et al., 2010].

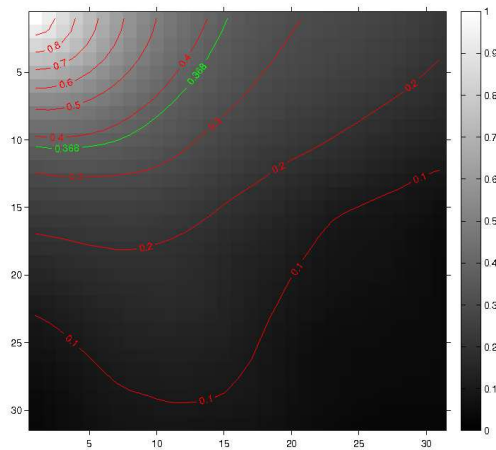
La rugosité intervient aussi dans l’étude des sols agricoles, notamment pour connaître l’érosion et l’humidité du sol, pour lequel il est dit dans [Roose, 1996] que



(a)



(b)



(c)

FIGURE 4.1.5 – (a) Image de profondeur par rapport au plan moyen ($Sq=1.12\text{cm}$), (b) image représentant le fonction d’auto-corrélation le long des deux axes, x et y, du nuage de galets, avec en vert la courbe correspondant à la valeur $1/e$

la rugosité influe sur le stock d’eau dans les sillons si la pente est inférieure à 30%.

Mais le domaine où la rugosité est primordiale reste l’étude de la rétro-diffusion radar. En effet, il est prouvé par Ulaby dans [Ulaby et al., 1979] que les paramètres de rugosité interviennent dans la réponse spectrale d’un signal radar. C’est-à-dire qu’un signal radar produira un écho si le terrain est rugueux pour la longueur d’onde. Ainsi,

selon [Girard and GIRARD, 1999], un terrain est lisse si :

$$S_q < \frac{\lambda}{8} \sin \alpha \quad (4.6)$$

et rugueux sinon, avec λ la longueur d'onde du signal radar et α l'angle d'incidence du signal.

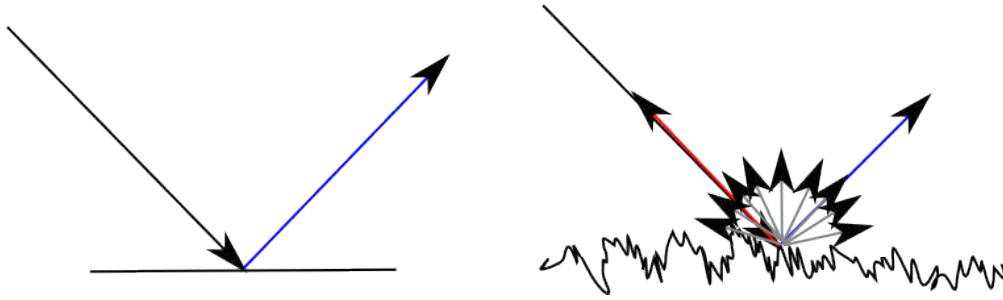


FIGURE 4.1.6 – Schéma du comportement de l'onde radar dans le cas lisse et rugueux, en bleue la direction spéculaire et en rouge la direction du hot-spot correspondant au signal rétro-diffusé.

Il existe une forte corrélation entre l'écart-type des hauteurs et le coefficient de rétro-diffusion radar, montré dans la figure 4.1.7 tiré de [Ulaby et al., 1979]. Il a aussi prouvé que le coefficient de rétro-diffusion radar semble parfaitement corrélé à la longueur d'auto-corrélation comme il est possible de le voir dans la figure 4.1.8, avec une figure tirée de l'article de De Keyser [De Keyser et al., 2010].

4.1.3 Discussions et perspectives

Nous avons trouvé, un certain nombre de paramètres de rugosité. Pour la longueur et la fonction d'auto-corrélation, comme pour l'écart type des hauteurs, la précision des valeurs dépend bien évidemment de la précision de la reconstruction discutée dans le chapitre précédent. Mais il est utile de se servir de l'état de l'art pour vérifier la cohérence des résultats.

En effet, selon Ulaby [Ulaby et al., 1979], pour que les paramètres de rugosité aient un sens, l'intervalle de discrétisation entre deux points doit être inférieur à un dixième de la longueur de corrélation. La longueur calculée dans notre expérience sur des galets est entre 10.5 et 15.5 pixels. Or, le pas est d'1 pixel donc inférieur au dixième de la plus petite longueur de corrélation. La valeur extraite a donc une meilleure signification. Même si la valeur de longueur de corrélation dépend très fortement de la valeur de longueur de transect [Bryant et al., 2007] et que son interprétation reste difficile [Davidson et al., 2003], elle reste un paramètre essentiel à la paramétrisation du signal radar [Ulaby et al., 1979].

De même Oh et Kay [Oh and Kay, 1998] ont montré que la longueur du profil (qui

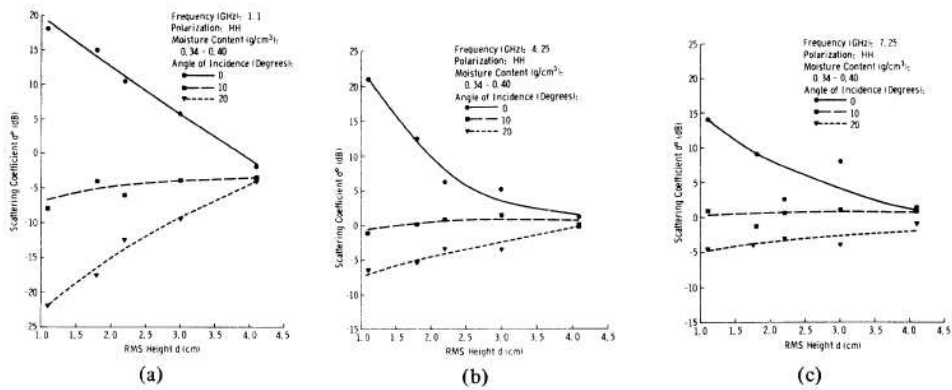


Fig. 6. Scattering coefficient dependence on rms height and angle of incidence at (a) 1.1 GHz, (b) 4.25 GHz, and (c) 7.25 GHz.

FIGURE 4.1.7 – Diagrammes de la rétro-diffusion radar en fonction du paramètre de rugosité définissant l'écart type des hauteurs (rms), pour différents angles d'incidence du signal radar. Les trois diagrammes représentent différentes fréquences de l'onde radar. (image tirée de l'article [Ulaby et al., 1979])

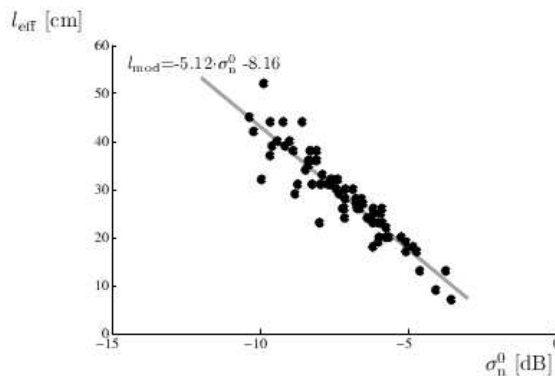


Figure 5: Dependence of the effective correlation length (l_{eff}) on the normalized backscatter coefficient (σ_n^0)

FIGURE 4.1.8 – Diagramme de la rétro-diffusion radar (σ_n^0) en fonction du paramètre de rugosité définissant la longueur d'auto-corrélation (l_{eff}) (image tirée de l'article [De Keyser et al., 2010]).

correspond sur la surface 3D à sa taille) doit être supérieure à 200 fois la longueur de corrélation. Or pour une longueur de corrélation de 15.5 pixel, une longueur estimée de 4000 pixels est prise en compte et qui est donc supérieure au $15.5 \times 200 = 3100$ pixels.

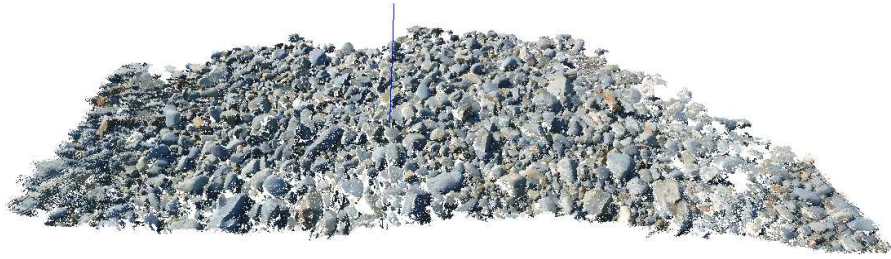


FIGURE 4.1.9 – Vue 3D des galets avec une valeur de S_q et S_a sur la surface de $S_a=1.23\text{cm}$ $S_q=1.12\text{cm}$

Des études récentes ont montré qu'il serait possible de croiser dendrométrie et paramètre de rugosité. En effet, ces études cherchent à montrer que la rugosité de la canopée pourrait servir à différencier les espèces. Nous avons donc choisi d'étudier une zone de feuillage, qui pour des raisons pratiques évidentes, a été étudiée sur un bonsaï (figure 4.1.10).

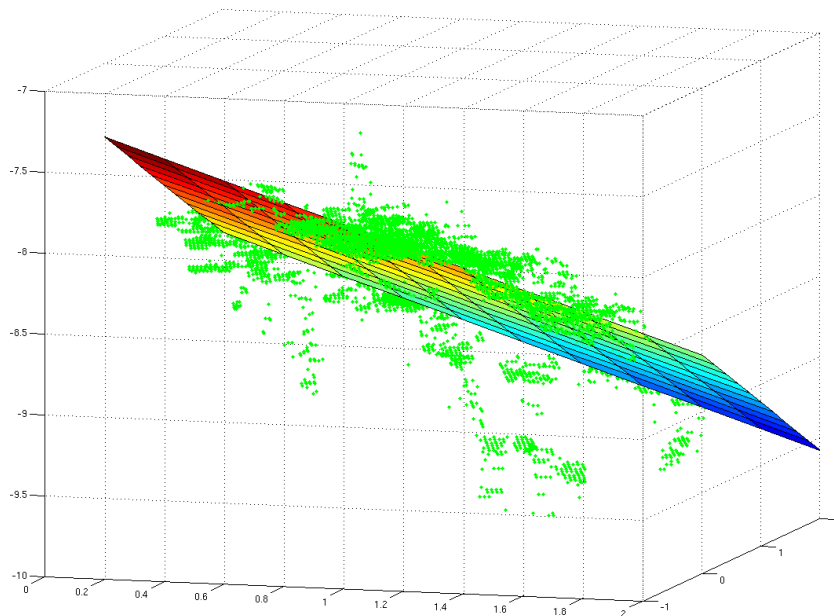


FIGURE 4.1.10 – Le feuillage du bonsaï et le plan moyen trouvé

4.2 Paramètres dendrométriques

La dendrométrie représente la caractérisation et les mesures physiques qu'il est possible d'effectuer sur la végétation. Ces paramètres sont variés avec par exemple la hauteur ou la forme des arbres mais aussi, des mesures statistiques comme la répartition ou la densité de peuplement. Si ces mesures peuvent se révéler complexes, de nombreux paramètres décrivant la forme des arbres sont définis à partir de paramètres simples et peuvent, a priori, être calculés sur les nuages de points 3D.

Les mesures pour l'instant s'effectuent *in-situ*, c'est-à-dire que les experts doivent se déplacer sur le terrain avec de multiples appareils de mesure pour calculer les différents paramètres. Ces techniques sont coûteuses aussi bien en temps qu'en argent à cause de la variété des appareils de mesures. Nous proposons donc de remplacer ces différents appareils de mesures par un simple appareil photo.

Malheureusement, il est bien connu qu'en stéréo-vision, la reconstruction de végétaux est un véritable défi. Des études récentes ont été effectuées pour la reconstruction de végétaux à partir de paramètres réalistes (en opposition aux algorithmes de création d'arbres purement génériques), notamment en se servant de paramètres dendrométriques et allométriques (paramètres définissant notamment la croissance des végétaux) comme dans les articles [Neubert et al., 2007] ou [Xu et al., 2007]. Dans l'article de Xu, [Xu et al., 2007], ce dernier utilise comme base des points 3D d'une plante en laboratoire obtenus par scanner laser. En condition réelle de mesure, c'est-à-dire à l'extérieur, cela semble être encore un défi.

Le but final de ces approches est la génération d'arbres réalistes et non de reconstruire des arbres réels. En effet, ces techniques utilisent pour la plupart une simple reconstruction du tronc puis rajoutent des feuilles au bout de branches engendrées comme dans les articles [Quan et al., 2006] ou [Tan et al., 2007].

Ainsi, notre approche reste originale puisqu'elle tente de reproduire fidèlement la réalité. Nous montrerons ainsi que nous obtenons des résultats exploitables malgré des erreurs plus présentes sur ce type de scènes que sur des galets.

Pour l'instant nous avons choisi de ne calculer que des paramètres dendrométriques simples. Nous laissons le soin aux spécialistes de choisir la pertinence des mesures complexes pouvant être effectuées sur les nuages 3D. Par exemple, une mesure comme la surface terrière d'un arbre dépend des valeurs de circonférence à 1m50 et se calcule à partir de différentes mesures d'épaisseurs des arbres. C'est pourquoi nous avons choisi de centrer notre travail sur le calcul des paramètres de grosseur des arbres et de hauteur, en particulier sur les dimensions du tronc et des feuilles. Mais nous ne pouvons savoir si ces mesures sont pertinentes ou pas, par contre nous pouvons prouver que la chaîne de traitement est capable de produire un nuage sur lequel, il est possible de mesurer l'épaisseur à un mètre cinquante.

Nous traiterons donc dans cette partie des paramètres de longueurs et d'épaisseurs.

4.2.1 Paramètres de longueurs

Il existe trois mesures classiques de hauteur :

- la hauteur totale ;
- la hauteur de "bois fort", correspondant à la hauteur du pied de l'arbre au niveau de la première branche ayant un diamètre supérieur ou égale à 7 cm ;
- la hauteur de "bois d'œuvre", correspondant à la hauteur du pied à la première "grosse" branche ;

Évidemment dans un soucis d'automatisation maximum, nous n'avons choisi de ne calculer que la hauteur totale. Il existe différents types d'appareils appelés hypsomètres capables de calculer la hauteur totale des arbres :

- des instruments électroniques très précis mais très chers et qui sont, en raison de leurs prix, peu utilisés.
- des informations de hauteur calculées à partir de nuages de points obtenus par LIDAR aéroportés [Nilsson, 1996], parfois en utilisant l'information onde-complète ou "full-wave form" [Heinzel and Koch, 2010]. L'utilisation de l'information onde-complète permet d'extraire de nombreux paramètres forestiers [Chauve et al., 2007]. Mais l'utilisation d'un lidar-aéroporté minimise aussi la possibilité d'utilisation, le matériel étant lourd et nécessitant un ULM ou un avion. Le coût reste un frein à son utilisation à grande échelle, même si les résultats obtenus permettent une plus grande précision qu'avec notre méthode (figure 4.2.1).
- des moyens rudimentaires comme d'une croix de bûcheron comme expliqué dans la figure 4.2.2 et qui restent les moyens les plus utilisés par les spécialistes.

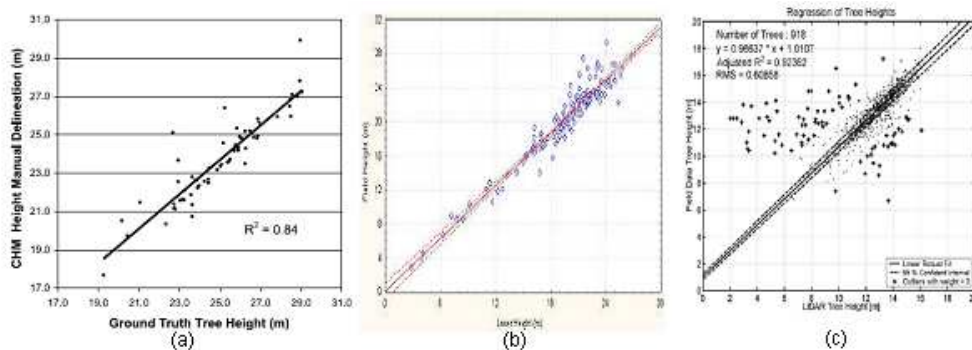


FIGURE 4.2.1 – Régression linéaire entre des hauteurs d'arbres mesurées sur le terrain et les hauteurs mesurées par le LIDAR aéroporté avec (a) $R^2 = 0,84$ ([Leckie et al., 2003]); (b) $R^2 = 0,92$ ([Lim et al., 2001]); (c) $R^2 = 0,92$ ([Morsdorf et al., 2004]) prouvant la précision des hauteurs mesurées par LIDAR aéroporté.

Il est difficile de prendre des photos d'arbres en pied, nous avons choisi de montrer un exemple de résultats obtenus sur un bonsaï en laboratoire.

Dans le cas des mesures manuelles effectuées sur le bonsaï 4.2.3, celles-ci ont été réalisées à l'aide d'un pied à coulisse. On estime donc la précision des mesures à celle du millimètre. Ainsi dans le cas de la hauteur totale, une valeur de 13,5 cm a été trouvée, nous donnant un peu plus de 3 millimètres d'erreur par rapport à la valeur

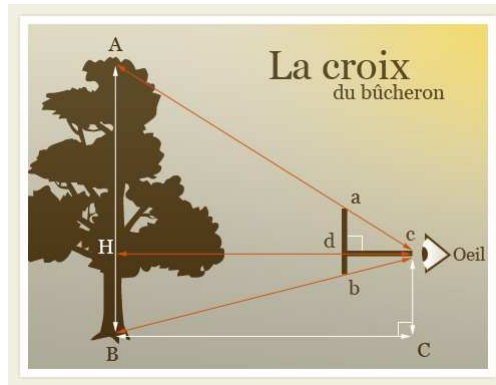


FIGURE 4.2.2 – Schéma explicatif de la croix du bûcheron, par le théorème des triangles semblables, la hauteur de l’arbre est égale à celle de la distance de la personne à l’arbre. La croix du bûcheron consiste en deux morceaux de bois de mêmes longueurs. En les mettant de manière orthogonale, il est possible d’utiliser le théorème de Thalès pour trouver la hauteur de l’arbre. En sachant que si $ab = cd$ alors $AB = BC$ puisque $\frac{ab}{AB} = \frac{cd}{BC}$.



FIGURE 4.2.3 – Calcul de hauteur à partir de mesures effectuées sur le nuage 3D du bonsaï

calculée sur le nuage de points 3D qui est de 13,2 cm. Pour le tronc une valeur de 11,1 cm a été calculée, nous donnant cette fois-ci 4 millimètres d’erreur.

En plus de ces résultats obtenus en laboratoire, nous avons expérimenté nos techniques sur des scènes d’arbres en extérieur. Or, le problème des arbres réside dans leur taille qui empêche de faire des photos de l’arbre en pied. Nous avons cherché et trouvé dans le bois de Vincennes des arbres capables d’être reconstruits en entier. Malheureusement, les filtrages successifs nous font perdre beaucoup de points du feuillage (figure 4.2.4).

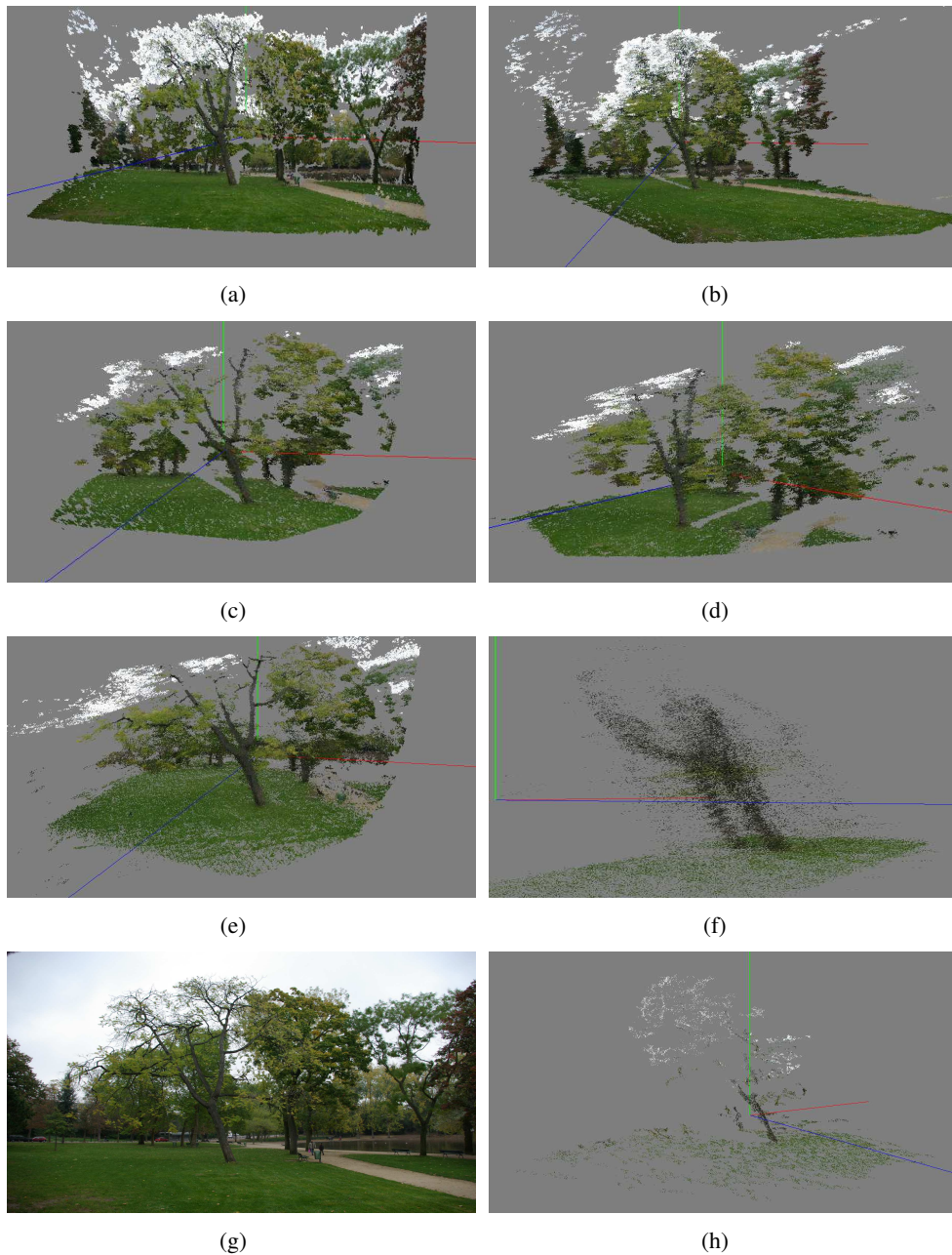


FIGURE 4.2.4 – (a)(b) Reconstruction 3D brute d'un seul couple d'images, (c)(d) reconstruction 3D filtrée du même couple, (e) nuage sans redondance avant lissage du nuage 3D, on s'aperçoit que de nombreux éléments importants ont disparu, (f) ici nous montrons le problème de reconstruction du tronc avec un dédoublement lié au manque de précision de la reconstruction, (g) un exemple d'une des 20 images de l'arbre, (h) reconstruction obtenue par l'algorithme de Furukawa et servant de comparaison.

D'autres mesures plus précises ont été effectuées sur les feuilles dans l'image 4.2.6, la mesure en bleue est celle qui nous a permis de déterminer le facteur d'échelle

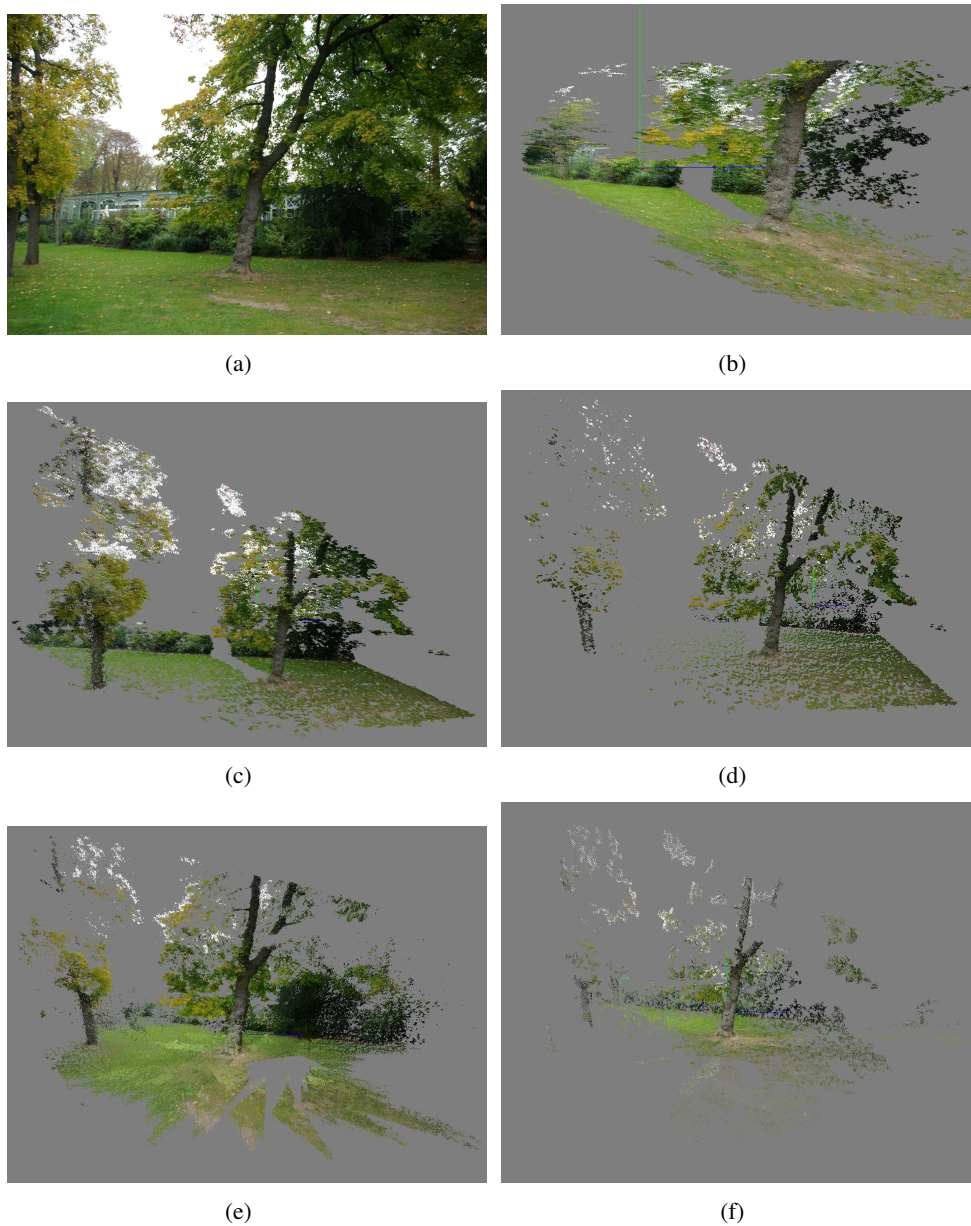


FIGURE 4.2.5 – (a) Exemple d'une des 114 images de l'arbre, (b)(c) reconstruction 3D brute de deux couples d'images différents, (d) reconstruction 3D filtrée du deuxième couple, (e) nuage sans redondance avec lissage du nuage 3D, (f) nuage de point reconstruit par l'algorithme de Furukawa.

entre scène 3D et centimètres. Les valeurs marquées sur l'image sont indicatives, et l'on trouvera dans les graphiques, à la figure 4.2.7, les valeurs effectivement mesurées, lors de l'analyse "a posteriori" en laboratoire. On remarque ainsi que l'erreur ne dépasse pas 2 cm, l'erreur moyenne est de 4,5 millimètre et l'erreur médiane est quant à elle de 3 mm. Ainsi les mesures effectuées sont de bonne qualité malgré l'erreur ajoutée par le choix difficile des pixels qui caractérisent le mieux la longueur.

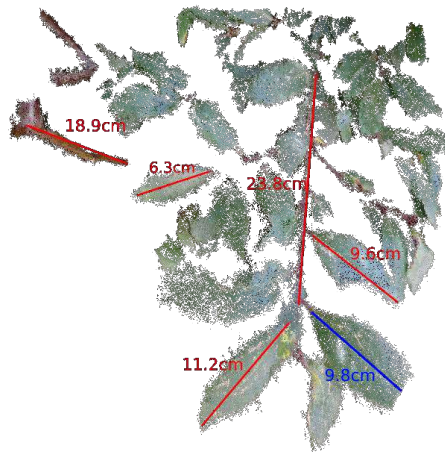


FIGURE 4.2.6 – Image des longueurs mesurées sur le nuage 3D de feuilles avec en bleu, la longueur servant au calcul du facteur d'échelle entre l'espace métrique et euclidien.

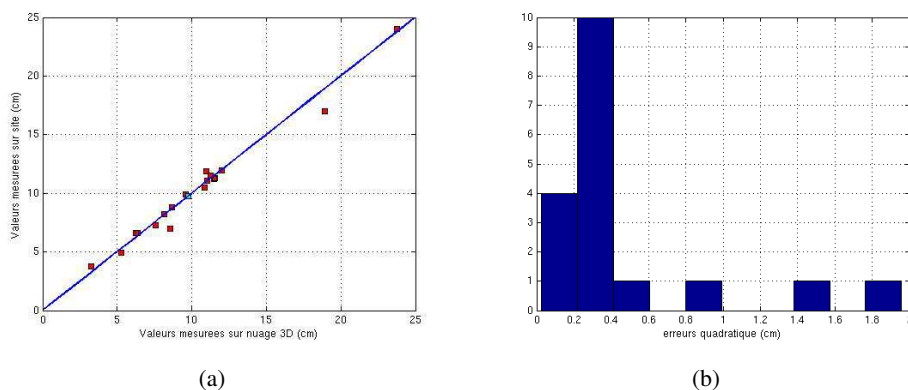


FIGURE 4.2.7 – (a) Rapport de l'erreur entre les valeurs des hauteurs 3D mesurées sur les valeurs réelles mesurées, (b) l'histogramme des erreurs en centimètres

4.2.2 Paramètres d'épaisseur

Pour les valeurs d'épaisseur, il existe trois mesures principales :

- le diamètre du tronc à 1m30 de hauteur à partir du pied de l'arbre, généralement calculé comme $d = \frac{d_{min} + d_{max}}{2}$ avec d le diamètre, d_{max} le diamètre maximum et d_{min} le diamètre minimum ;
- la circonférence du tronc à 1m30 de hauteur ;
- la surface terriale correspondant à la surface de la section d'arbre à 1m30, qui est calculée telle que :

$$g = \frac{c^2}{4.\pi} \quad (4.7)$$

avec g surface terrière en m^2 et c la circonférence de l'arbre en mètre.

Le problème est que ces différentes mesures sont faites avec des instruments malcommodes :

- pour le diamètre, les mesures sont faites au pied à coulisse ;
- pour la circonférence, un mètre ruban est utilisé ;

or, ces mesures sont longues et nécessitent de très nombreuses heures de travail, nous proposons donc une nouvelle manière de faire ces mesures.

Pour faire toutes ces mesures, il suffit de déterminer la hauteur d'un plan de coupe, et de prendre en compte les points qui se trouvent proches de ce plan. A partir de ce plan de coupe, nous estimons l'ellipsoïde s'adaptant au mieux aux points 3D, projetés sur le plan, qui nous permet d'extraire le diamètre minimum, maximum et une valeur approchée de la circonférence.

Pour trouver l'ellipsoïde qui s'attache le mieux aux données, il faut trouver tous les points 3D se situant à 1m30. Pour cela, nous choisissons manuellement la hauteur, et le plan correspondant est choisi. Tous les points qui sont proches de ce plan sont sélectionnés comme vu sur la figure 4.9(a). Ces points forment ainsi la figure que nous pouvons voir en 4.9(b) et sur laquelle nous pouvons trouver les paramètres de l'ellipsoïde définissant les valeurs d'épaisseur de l'arbre.

La figure 4.9(b) montrent les paramètres de l'ellipsoïde estimés. Ainsi les mesures trouvées sont : le demi-grand axe $a = 47.7\text{cm}$ et le demi-petit axe $b = 45.8\text{cm}$. Le périmètre de l'arbre calculé est de $c = 293.8\text{cm}$. Or, les mesures faites manuellement ont montré que le périmètre de l'arbre est de 3m, mesures faites à l'aide d'un mètre enrouleur. Cette erreur de 6cm est liée à différents facteurs comme la non fermeture du tronc (le tronc ne se ferme pas parfaitement) ou les différents biais de mesures :

- l'estimation de l'ellipsoïde ;
- le calcul du périmètre qui se fait par approximation de l'intégrale elliptique ;
- les mesures faites sur le terrain ;
- le choix des mesures semblables pour le calcul du facteur d'échelle.

A partir des données précédentes, nous avons calculé la surface terrière de l'arbre qui est de $g = 0.687 m^2$.

4.3 Perspectives

De très nombreux autres paramètres peuvent être extraits des nuages de points 3D représentant des arbres. Il reste par exemple à extraire des paramètres de forme, caractérisant la forme d'un arbre ou d'une canopée. Elles sont définies par un certain nombre de coefficients :

- le coefficient de décroissance, représentant le rapport entre le diamètre à mi-hauteur de la tige sur le diamètre à 1m30 $k = \frac{d_H}{d}$;
- le coefficient de réduction, défini par $r = 1 - k$;
- le défilement ou décroissance métrique négative, défini par $d_{mm} = \frac{d - d_H}{\frac{H}{2} - H_{1.3m}}$ avec $H_{1.3m}$ la hauteur entre la section de culée et 1m30 ;
- le coefficient de forme, correspondant au rapport entre le volume réel et le volume d'un cylindre de base g et de hauteur H , défini de tel sorte que $f = \frac{V}{g.H}$ avec

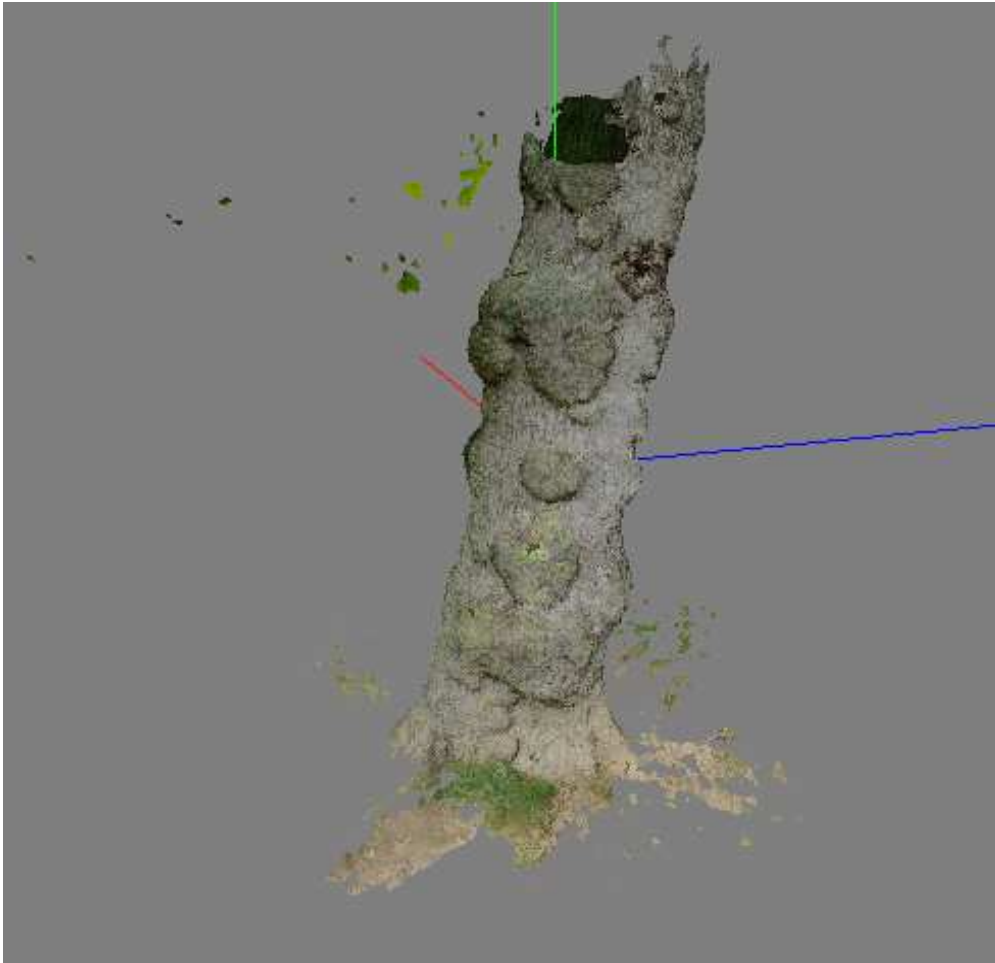


FIGURE 4.2.8 – Scène 3D d'un tronc, obtenu en sélectionnant manuellement le tronc sur la figure 4.2.5.

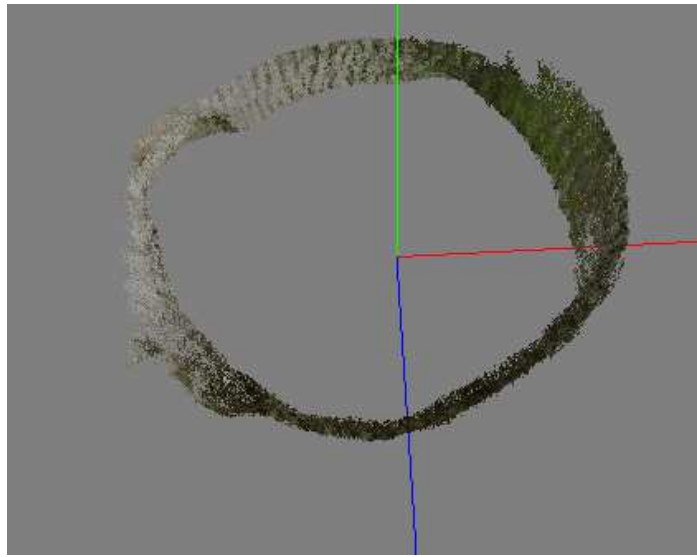
v le volume de l'arbre.

Ce travail devra se poursuivre en collaboration avec les utilisateurs de tels paramètres, et avec eux définir de nouveaux paramètres de surface à extraire sur le nuage 3D. Nous espérons donc que de nouvelles mesures verront le jour dans le futur.

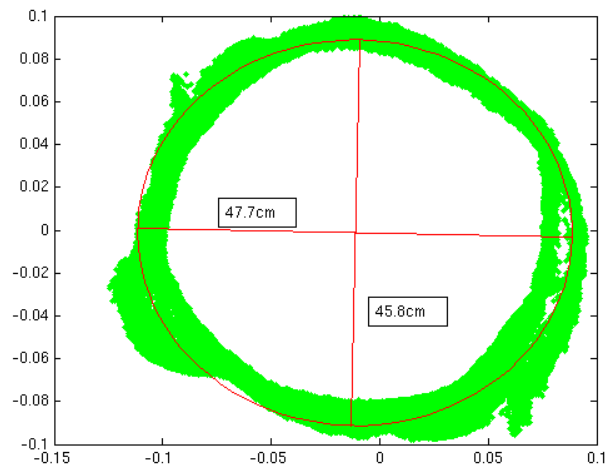
4.3.1 Utilisation d'un quadri-coptère

Nous avons notamment choisi de travailler sur la végétation en collaboration avec l'ESIEA, ceux-ci possédant un drone quadri-coptère capable de supporter le poids d'un appareil photo. L'utilisation d'un tel drone permet notamment l'acquisition d'images à plus haute altitude et permet ainsi de mettre en exergue d'autres types d'échelles de paramètres de rugosité. Nous avons notamment étudié la végétation sur de la lande en forêt de Fontainebleau [Gademer et al.,].

L'utilisation d'un drone quadri-coptère pose tout de même de nombreux prob-



(a)



(b)

FIGURE 4.2.9 – (a) morceau de tronc correspondant au plan de coupe. On choisit les points les plus proches du plan ; le plan de coupe étant sélectionné manuellement à la hauteur d'1m30, (b) à partir de ces points l'ellipsoïde est estimé avec ses paramètres.

lèmes pour l'obtention de photos dans les conditions optimales. Notamment le poids trop faible de l'appareil volant implique de nombreuses vibrations et mouvements qui rendent les images très souvent floues. Ainsi, une campagne de validation de la méthode fut effectuée sur un parterre de végétation devant l'école ESIEA (figure 4.3.1).

Lorsqu'on imagine toutes les utilisations possibles des paramètres de dendrométrie, cette nouvelle méthodologie de mesure pouvant être obtenue dans toutes

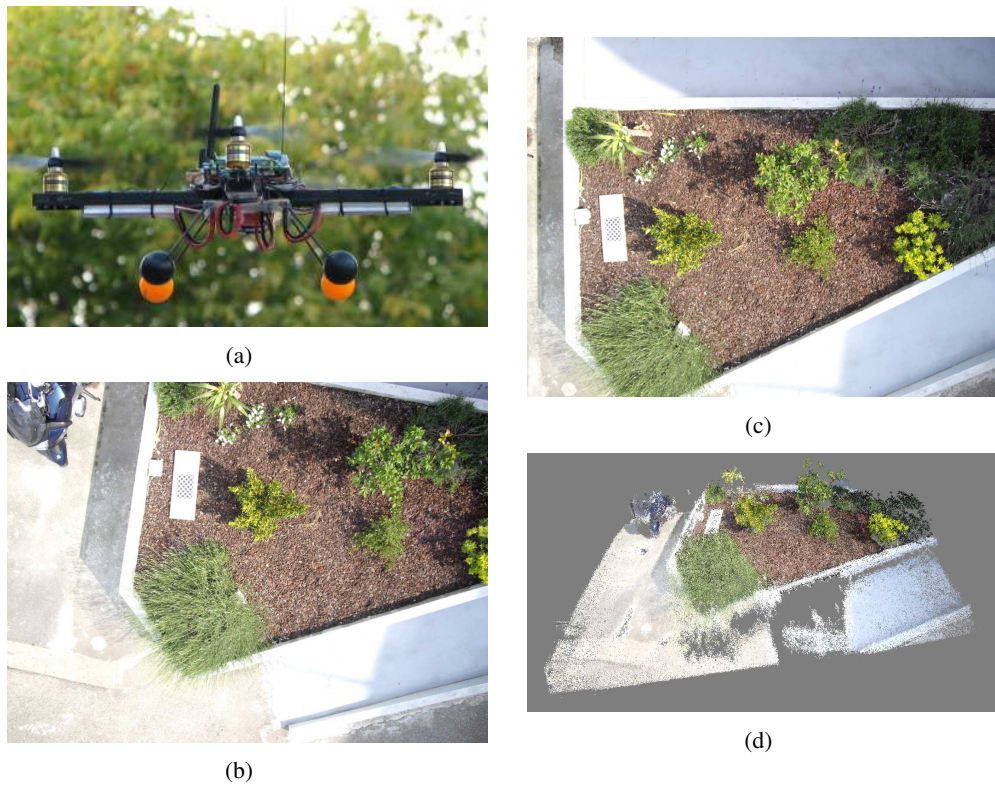


FIGURE 4.3.1 – (a) Image du drone quadri-coptère conçu et réalisé par l'équipe de recherche de l'ESIEA, (b)(c) Exemples de photos obtenues par le drone sur un parterre de plantes, (d) exemple de reconstruction 3D obtenue.

les conditions possibles et à bas coût, offre de nouvelles perspectives dans de nombreux domaines. Même si aujourd'hui des méthodes existent et offrent peut-être de meilleurs résultats, de nombreuses études effectuées dans des régions difficiles empêchent l'utilisation de ce type d'appareil. Ainsi, il est ici proposé une offre alternative capable d'offrir des valeurs numériques quantifiables avec un simple appareil photo.

Conclusion

Dans ce chapitre, nous avons abordé deux domaines très différents pour lesquels l'extraction des mesures n'est pas simple. Pourtant il semble que leurs utilités ne soient plus à démontrer et que les différents chercheurs travaillant sur le sujet aient du mal à extraire cette matière première que sont les mesures. Nous pouvons donc, aux vues des résultats obtenus, dire que les mesures apportent une nouvelle perspective pour les chercheurs. Malgré tout, des améliorations peuvent être apportées, notamment dans le domaine dendrométrique. Un partenariat avec le muséum national d'histoire naturelle nous permettra sûrement d'extraire des paramètres plus complexes et d'améliorer la

qualité des mesures existantes.

En effet, nous nous sommes contentés de mesures physiques sans chercher à extraire des mesures morphologiques très utiles en botaniques notamment pour la classification des espèces. Nous souhaiterions donc développer au maximum le nombre de mesures extractibles sur les nuages et spécifier certaines contraintes pour la botanique et l'écologie.

Conclusion et perspectives

Le but de cette thèse était l'utilisation de techniques récentes de reconstruction 3D pour l'extraction de paramètres de surfaces. Ces travaux trouvaient leur intérêt dans la difficulté pour de nombreux spécialistes d'extraire des informations quantitatives sur des surfaces.

Nous avons choisi une approche pratique du problème, c'est à dire que tout les choix faits au cours de cette thèse furent motivés par la problématique d'être utilisé par des non-spécialistes de la reconstruction 3D dans n'importe quelle condition.

Ces contraintes nous ont menés à utiliser des méthodes robustes aux problèmes de scènes comprenant des objets mobiles et qui nous ont permis de créer des nuages de points 3D. Malheureusement ceux-ci souffrant d'un certain nombre de problèmes, il était impossible d'y effectuer des mesures. Nous avons donc du trouver un moyen de filtrer ces nuages pour nous permettre d'y extraire les paramètres de surfaces.

Dans l'optique de vérifier les mesures, nous avons envisagé un cas d'étude nous permettant de comparer les mesures effectuées sur le nuage avec des mesures faites au pied à coulisse. Les résultats, bien qu'indicatifs, nous ont permis de donner un ordre de grandeur des erreurs de mesures.

Enfin, nous avons pu extraire des informations quantitatives des nuages reconstruits. Ces mesures, faites dans le cadre de partenariats, nous ont permis de tester leur utilité en condition réelle d'utilisation.

Nous avons abordé dans ce manuscrit toute la chaîne de traitements 3D nous permettant d'obtenir, à partir d'une série de photos prises sans contraintes, un nuage de points 3D sur lequel des mesures ont pu être effectuées. Pour chaque algorithme, une étude de la bibliographie nous a permis de choisir l'algorithme le plus adapté à notre problématique. Ces algorithmes devaient être robustes, rapides et peu gourmands en ressources mémoire.

Nous avons choisi dans ce manuscrit d'aborder tout ces algorithmes selon deux angles différents, un angle orienté vers la vision, se concentrant sur la production de contenus 3D à partir d'une collection d'images, et un angle orienté informatique graphique, traitant des aspects du traitement de l'information 3D.

Nous avons tenté d'avoir un regard critique sur notre travail en mettant en avant les qualités et défauts des algorithmes choisis tout en expliquant les raisons de nos choix.

Pour contribuer à ce regard critique, nous avons cherché à vérifier la cohérence des mesures effectuées et à discuter des sources possibles des erreurs. A partir de ce travail nous avons pu extraire un certain nombre de paramètres sur les nuages 3D.

Ces paramètres représentent bien les possibilités de la chaîne de traitement, avec la rugosité de terrain servant aussi bien à l'étude du signal radar que l'érosion des terrains agricoles et la dendrométrie expliquant le comportement des arbres dans le but d'étudier la bio-diversité.

Extraire des paramètres sur une surface 3D n'était pas simple, surtout à partir de photos prises sans contraintes. Toute la difficulté de cette thèse résidait dans l'idée de n'imposer qu'un minimum de contraintes physiques et financières, pour offrir un nouvel outil de mesures biologiques et géo-physiques au plus grand nombre d'utilisateurs possibles.

Nous avons atteint ce but en créant un outil capable, à partir d'un simple appareil photo du commerce, d'extraire des paramètres de rugosité ou dendrométrique. Le nombre d'exemples traités ainsi que les nombreuses mesures effectuées ont permis de confirmer le bon fonctionnement des outils et de la méthode.

L'outil développé permet de faire des mesures sur des nuages 3D reconstruits à partir d'images prises avec peu de contraintes. Seule des contraintes sur la qualité des photos et la manière de faire les acquisitions permettent de garantir la bonne qualité des reconstructions et de l'auto-calibrage et par voie de conséquence des mesures. Il aurait été intéressant d'étudier l'impact de la qualité des capteurs CCD des appareils photos sur la qualité des mesures effectuées.

Nos travaux ont amené à la publication de quatre articles de conférence avec comité de relecture et d'un article dans une conférence francophone d'écologie (à monastir en tunisie) sans publication. Le côté pluri-disciplinaire de cette thèse a permis la publication de trois articles en télédétection et d'un article dans une conférence sur l'écologie. Sur ces quatre articles, trois ont été publiés dans des conférences internationales (dont une francophone) et une dans une conférence française.

La liste des articles sont :

- *Roughness measurement from multi-stereo reconstruction*
Petitpas, B and Beaudoin, L and Roux, M and Rudant, JP.
ISPRS Symposium on Photogrammetric Computer Vision and Image Analysis
Saint-Mandé, France, 2010
- *Developing a lowcost Vertical Take Off and Landing Unmanned Aerial System for centimetric monitoring of biodiversity the Fontainebleau Forest case*
Gademer, A. and Petitpas, B. and Mobaied, S. and Beaudoin, L. and Riera, B. and Roux, M. and Rudant, JP
Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International
- *Using centimetric visible imagery obtained from an UAV quadrotor for classification of ERS images*
A. Gademer, B. Petitpas, L. Beaudoin, T. Tanzi, B. Riera, J.P. Rudant
European Space Agency Living Planet Symposium, 2010

- *Nouvelles méthodologies de mesures sur la végétation à partir de reconstructions 3D obtenues à partir de séquences photographiques terrestres ou aériennes*

Benoît Petitpas, Antoine Gademer, Laurent Beaudoin, Michel Roux, Bernard Riera, Jean-Paul Rudant

XIème Journées Scientifiques du Réseau Télédétection de l'Agence Universitaire de la Francophonie

4.3.2 Perspectives

Les contributions nous renseignent quand au caractère transverse du sujet. Ce point et les différents développements, longuement menés, n'ont pas permis d'approfondir certains points qui auraient mérité un travail plus important pour améliorer les résultats, notamment certains points mis en lumière au cours du manuscrit tel que la sensibilité de la chaîne de traitement à l'auto-calibrage. Un travail plus approfondi sur ce sujet aurait permis d'améliorer les résultats obtenus significativement. De plus, les tests effectués ne sont pas suffisants pour mesurer l'exactitude de la méthode et une campagne d'obtention d'une vérité terrain aurait permis de surmonter cette difficulté. Toutefois, les outils développés ainsi que la robustesse de la chaîne nous permet d'envisager différentes applications de ces travaux. En effet, la généralité et la robustesse des algorithmes utilisés permettent d'ensaisiner la reconstruction de n'importe quel type de scènes. Ainsi de nombreuses applications pourraient être dérivées de la chaîne développée. Or, pour que ces travaux soient utilisés dans le futur, il était nécessaire de simplifier la production du nuage de points 3D et permettre des mesures hétérogènes.

Dans les perspectives de développement de nouvelles mesures, il faut mettre en avant la collaboration prometteuse avec l'esiea via l'utilisation d'images d'un drone quadri-coptère offrant un nouvel espace de mesure avec la possibilité de reconstruire de la 3D à partir d'images prises à 20 mètres de hauteur. De plus, une nouvelle équipe de cette même école travaille actuellement sur un sous-marin radio-commandé (ROV) capable de cartographier les fonds marins. Donner une dimension 3D à cette cartographie serait un nouveau défi même si des travaux ont déjà été effectués sur la stéréo-vision sous-marine ([Brandou, 2008]).

Nous n'avons malheureusement pas eu le temps d'explorer toutes les pistes possibles concernant l'obtention de paramètres sur des surfaces. Malgré l'envie de faire la chaîne de traitement la plus générique possible, nous aurions aimé contacter plus de personnes intéressées par le sujet afin de spécialiser la chaîne pour d'autres types de paramètres.

Une autre perspective est la mise en évidence des phénomènes de rétro-diffusion radar avec la rugosité de surfaces faites sur des surfaces et non des profils. Nous avons le projet de réaliser une campagne de comparaison sur le calcul de la rugosité dans la baie du mont Saint-Michel.

4.3.3 MobyDick

Les perspectives pour la rugosité sont d'ouvrir au plus grand nombre de chercheurs la possibilité d'effectuer des mesures à partir d'images prises sans contraintes. Pour cela, la publication de codes sources clairs avec des interfaces graphiques permettrait de pérenniser cette méthode de mesure. C'est ce qui sera fait avec la publication du logiciel MobyDick, développé au cours de cette thèse, avec un manuel d'utilisation et une documentation qui reste à écrire.

En effet, au cours de cette thèse fut produit un certain nombre de codes et de programmes regroupés au sein d'un seul programme appelé "MobyDick" en hommage au livre d'Herman Melville permettant de faire les mesures. Cette outil contient :

- Un utilitaire permettant la reconstruction 3D à partir de photos comprenant lui-même :
 - Un appareilleur de points saillants
 - Un rectificateur épipolaire de couple d'images
 - Un algorithme de mise en correspondance dense
 - Un algorithme de triangulation
- Un utilitaire de filtrage 3D comprenant
 - Un supprimeur de points isolés
 - Un supprimeur de points redondants
 - Un lisseur
- Un visualisateur permettant
 - des mesures de longueurs
 - d'extraire l'écart type des hauteurs à partir d'un plan moyen
 - de produire une carte de profondeur, sur laquelle sera effectuée une fonction d'auto-corrélation
 - d'extraire un plan de coupe à n'importe quelle hauteur

Ce programme résume à lui seul tout le travail fourni durant ces trois années de thèse.

Nous avons pensé à de nombreuses suites pour ce logiciel comme tenter d'améliorer les performances de nos programmes qui restent lents. En effet aucune tâche n'a été parallélisée alors que de nombreuses parties auraient pu l'être. Sa mise en ligne gratuite et open-source permettra, je l'espère, de pérenniser ce travail. MobyDick est basé sur une bibliothèque de fonctions nommée OpenCV qui a la chance d'être extrêmement maintenue à jour.

Bibliographie

- [Adamson and Alexa, 2003] Adamson, A. and Alexa, M. (2003). Approximating and intersecting surfaces from points. In *Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 230–239. Eurographics Association.
- [Adamson and Alexa, 2004] Adamson, A. and Alexa, M. (2004). Approximating bounded, non-orientable surfaces from points (figures 5, 6, and 7). In *SMI*, page 393.
- [Alexa and Adamson, 2004] Alexa, M. and Adamson, A. (2004). On normals and projection operators for surfaces defined by point sets. In *Proceedings of Symposium on Point-Based Graphics*, volume 4, pages 149–155. Citeseer.
- [Alexa et al., 2001] Alexa, M., Behr, J., Cohen-Or, D., Fleishman, S., Levin, D., and Silva, C. (2001). Point set surfaces. In *Proceedings of the Conference on Visualization '01*, pages 21–28. IEEE Computer Society.
- [Amenta et al., 1998] Amenta, N., Bern, M., and Kamvysselis, M. (1998). A new voronoi-based surface reconstruction algorithm. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 415–421. ACM.
- [Amenta and Kil, 2004] Amenta, N. and Kil, Y. J. (2004). Defining point set surfaces. In *ACM*, volume 23 of *ACM Transactions on Graphics (TOG)*, pages 264–270.
- [Arya and Mount, 1993] Arya, S. and Mount, D. (1993). Algorithms for fast vector quantization. In *Data Compression Conference, 1993. DCC'93.*, pages 381–390. IEEE.
- [Arya et al., 1998] Arya, S., Mount, D., Netanyahu, N., Silverman, R., and Wu, A. (1998). An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6) :891–923.
- [Aubrecht et al., 2010] Aubrecht, C., Hofle, B., Hollaus, M., Klotzl, M., Steinnocher, K., and Wagner, W. (2010). Vertical roughness mapping-based classification of the vertical vegetation structure in forested areas. In *Proceedings of ISPRS TC VII Symposium : 100 Years ISPRS*, pages 35–40.
- [Baghdadi et al., 2008] Baghdadi, N., Cerdan, O., Zribi, M., Auzet, V., Darboux, F., El Hajj, M., and Kheir, R. (2008). Operational performance of current synthetic aperture radar sensors in mapping soil surface characteristics in agricultural environments : application to hydrological and erosion modelling. *Hydrological Processes*, 22(1) :9–20.
- [Barazzetti et al., 2010] Barazzetti, L., Scaioni, M., and Remondino, F. (2010). Orientation and 3d modelling from markerless terrestrial images : combining accuracy with automation. *The Photogrammetric Record*, 25(132) :356–381.

- [Bay et al., 2008] Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3) :346–359.
- [Besl and McKay, 1992] Besl, P. and McKay, N. (1992). A method for registration of 3-d shapes. *IEEE Transactions on pattern analysis and machine intelligence*, pages 239–256.
- [Boykov et al., 1998] Boykov, Y., Veksler, O., and Zabih, R. (1998). Markov random fields with efficient approximations. In *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, pages 648–655. IEEE.
- [Brandou, 2008] Brandou, V. (2008). *Stéréovision locale et Reconstruction 3D/4D*. PhD thesis, Nice Sophia-Antipolis.
- [Brown and Lowe, 2005] Brown, M. and Lowe, D. (2005). Unsupervised 3D object recognition and reconstruction in unordered datasets. In *3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on*, pages 56–63. IEEE.
- [Bryant et al., 2007] Bryant, R., Moran, M., Thoma, D., Collins, C., Skirvin, S., Rahman, M., Slocum, K., Starks, P., Bosch, D., and Dugo, M. (2007). Measuring Surface Roughness Height to Parameterize Radar Backscatter Models for Retrieval of Surface Soil Moisture. *IEEE Geoscience and Remote Sensing Letters*, 4(1) :137.
- [Chauve et al., 2007] Chauve, A., Durrieu, S., Bretar, F., Pierrot Deseilligny, M., and Puech, W. (2007). Processing full-waveform lidar data to extract forest parameters and digital terrain model : Validation in an alpine coniferous forest.
- [Comaniciu and Meer, 2002] Comaniciu, D. and Meer, P. (2002). Mean shift : A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, pages 603–619.
- [Davidson et al., 2000] Davidson, M., Le Toan, T., Mattia, F., Satalino, G., Manninen, T., and Borgeaud, M. (2000). On the characterization of agricultural soil roughness for radar remote sensing studies. *IEEE Transactions on Geoscience and Remote Sensing*, 38(2) :630–640.
- [Davidson et al., 2003] Davidson, M., Mattia, F., Satalino, G., Verhoest, N., Le Toan, T., Borgeaud, M., Louis, J., and Attema, E. (2003). Joint statistical properties of RMS height and correlation length derived from multisite 1-m roughness measurements. *Geoscience and Remote Sensing, IEEE Transactions on*, 41(7) :1651–1658.
- [De Keyser et al., 2010] De Keyser, E., Lievens, H., Vernieuwe, H., Marzahn, P., Ludwig, R., de Lathauwer, E., Pauwels, V., de Baets, B., and Verhoest, N. (2010). Uncertainty assessment on sar retrieved soil moisture based on a regression model for roughness.
- [Dey and Goswami, 2003] Dey, T. and Goswami, S. (2003). Tight cocone : a watertight surface reconstructor. *Journal of Computing and Information Science in Engineering*, 3 :302.
- [Dorren and Heuvelink, 2004] Dorren, L. and Heuvelink, G. (2004). Effect of support size on the accuracy of a distributed rockfall model. *International Journal of Geographical Information Science*, 18(6) :595–609.
- [Engman and Chauhan, 1995] Engman, E. and Chauhan, N. (1995). Status of microwave soil moisture measurements with remote sensing. *Remote Sensing of Environment*, 51(1) :189–198.

- [Faugeras, 1993] Faugeras, O. (1993). *Three-dimensional computer vision : a geometric viewpoint*. MIT Press Ltd.
- [Faugeras et al., 1992] Faugeras, O., Luong, Q., and Maybank, S. (1992). Camera self-calibration : Theory and experiments. In *Computer Vision—ECCV’92*, pages 321–334. Springer.
- [Fischler and Bolles, 1981] Fischler, M. and Bolles, R. (1981). Random sample consensus : a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6) :381–395.
- [Fleishman et al., 2005] Fleishman, S., Cohen-Or, D., and Silva, C. (2005). Robust moving least-squares fitting with sharp features. *ACM Transactions on Graphics (TOG)*, 24(3) :544–552.
- [Franco and Boyer, 2005] Franco, J.-S. and Boyer, E. (2005). Fusion of Multi-View Silhouette Cues Using a Space Occupancy Grid. In *ICCV ’05 : Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1747–1753, Beijing, China. IEEE Computer Society.
- [Friedman et al., 1977] Friedman, J., Bentley, J., and Finkel, R. (1977). An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3) :209–226.
- [Furukawa and Ponce, 2009] Furukawa, Y. and Ponce, J. (2009). Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, pages 1362–1376.
- [Fusiello et al., 2000] Fusiello, A., Trucco, E., and Verri, A. (2000). A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1) :16–22.
- [Gademer et al.,] Gademer, A., Petitpas, B., Mobaied, S., Beaudoin, L., Riera, B., Roux, M., and Rudant, J. Developing a lowcost Vertical Take Off and Landing Unmanned Aerial System for centimetric monitoring of biodiversity the Fontainebleau Forest case. In *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*, pages 600–603. IEEE.
- [Gellert et al., 1975] Gellert, W., Kuestner, H., Hellwich, M., and Kaestner, H. (1975). Editors. the vnr concise encyclopaedia of mathematics.
- [Gerrits and Bekaert, 2006] Gerrits, M. and Bekaert, P. (2006). Local stereo matching with segmentation-based outlier rejection. In *Computer and Robot Vision, 2006. The 3rd Canadian Conference on*, page 66. IEEE.
- [Girard and GIRARD, 1999] Girard, M. and GIRARD, C. (1999). Traitement des données de télédétection, éditions dunod. Technical report, ISBN 2-1000-4185-1, 529 p.
- [Govers et al., 2000] Govers, G., Takken, I., and Helming, K. (2000). Soil roughness and overland flow. *Agronomie*, 20(2) :131–146.
- [Guennebaud and Gross, 2007] Guennebaud, G. and Gross, M. (2007). Algebraic point set surfaces. In *ACM SIGGRAPH 2007 papers*, pages 23–es. ACM.
- [Harris and Stephens, 1988] Harris, C. and Stephens, M. (1988). A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK.
- [Hartley and Sturm, 1997] Hartley, R. and Sturm, P. (1997). Triangulation. *Computer Vision and Image Understanding*, 68(2) :146–157.
- [Hartley and Zisserman, 2004] Hartley, R. and Zisserman, A. (2004). *Multiple view geometry in computer vision*. Cambridge Univ Pr, second edition.

- [Hartley and Zisserman, 2000] Hartley, R. I. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN : 0521623049.
- [Heinzel and Koch, 2010] Heinzel, J. and Koch, B. (2010). Exploring full-waveform lidar parameters for tree species classification. *International Journal of Applied Earth Observation and Geoinformation*.
- [Hernández Esteban and Schmitt, 2004] Hernández Esteban, C. and Schmitt, F. (2004). Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3) :367–392.
- [Hirschmuller et al., 2002] Hirschmuller, H., Innocent, P., and Garibaldi, J. (2002). Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47(1) :229–246.
- [Kazhdan et al., 2006] Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, pages 61–70. Eurographics Association.
- [Klaus et al., 2006] Klaus, A., Sormann, M., and Karner, K. (2006). Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. 3 :15–18.
- [Koller et al., 2004] Koller, D., Turitzin, M., Levoy, M., Tarini, M., Crocchia, G., Cignoni, P., and Scopigno, R. (2004). Protected interactive 3d graphics via remote rendering. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 695–703. ACM.
- [Kolmogorov and Zabih, 2001] Kolmogorov, V. and Zabih, R. (2001). Computing visual correspondence with occlusions using graph cuts.
- [Kolmogorov and Zabih, 2002] Kolmogorov, V. and Zabih, R. (2002). Multi-camera scene reconstruction via graph cuts. *Computer Vision—ECCV 2002*, pages 8–40.
- [Kruppa, 1913] Kruppa, E. (1913). Zur ermittlung eines objektes aus zwei perspektiven mit innerer orientierung, sitz. *Ber. Akad. Wiss., Wien, Math. Naturw. Kl., Abt. IIa*, 122 :1939–1948.
- [Lancaster and Salkauskas, 1981] Lancaster, P. and Salkauskas, K. (1981). Surfaces generated by moving least squares methods. *Mathematics of computation*, 37(155) :141–158.
- [Leckie et al., 2003] Leckie, D., Gougeon, F., Hill, D., Quinn, R., Armstrong, L., and Shreenan, R. (2003). Combined high-density lidar and multispectral imagery for individual tree crown analysis. *Canadian Journal of Remote Sensing*, 29(5) :633–649.
- [Leloglu et al., 1998] Leloglu, M., Roux, M., and Maître, H. (1998). Dense urban DEM with three or more high-resolution aerial images. In *ISPRS Symposium on GIS-Between Visions and Applications. Stuttgart, Germany*, pages 347–352.
- [Levin, 1998] Levin, D. (1998). The approximation power of moving least-squares. *Mathematics of computation*, 67(224) :1517–1532.
- [Lifshitz and Pizer, 1990] Lifshitz, L. and Pizer, S. (1990). A multiresolution hierarchical approach to image segmentation based on intensity extrema. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(6) :529–540.
- [Lim et al., 2001] Lim, K., Treitz, P., Groot, A., and St-Onge, B. (2001). Estimation of individual tree heights using lidar remote sensing. In *Proceedings of the Twenty-Third Annual Canadian Symposium on Remote Sensing, Quebec, QC*.
- [Longuet-Higgins, 1981] Longuet-Higgins, H. (1981). A computer program for reconstructing a scene from two projections. *Nature*, 293(11) :133–135.

- [Languet-Higgins, 1987] Languet-Higgins, H. (1987). A computer algorithm for reconstructing a scene from two projections. *Readings in computer vision : issues, problems, principles, and paradigms*, page 61.
- [Lorensen and Cline, 1987] Lorensen, W. and Cline, H. (1987). Marching cubes : A high resolution 3d surface construction algorithm. *ACM Siggraph Computer Graphics*, 21(4) :163–169.
- [Lourakis and Argyros, 2004] Lourakis, M. and Argyros, A. (2004). The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. *ICS/FORTH Technical Report TR*, 340.
- [Lowe, 1999] Lowe, D. (1999). Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, volume 2, pages 1150–1157.
- [Maneewongvatana and Mount, 2002] Maneewongvatana, S. and Mount, D. (2002). Analysis of approximate nearest neighbor searching with clustered point sets. *Data structures, near neighbor searches, and methodology : fifth and sixth DIMACS implementation challenges : papers related to the DIMACS challenge on dictionaries and priority queues (1995-1996) and the DIMACS challenge on near neighbor searches (1998-1999)*, 59 :105.
- [Margreth and Funk, 1999] Margreth, S. and Funk, M. (1999). Hazard mapping for ice and combined snow/ice avalanches—two case studies from the swiss and italian alps. *Cold Regions Science and Technology*, 30(1-3) :159–173.
- [Mcglone et al., 1980] Mcglone, C., Mikhail, E., and Bethel, J. (1980). Manual of photogrammetry. American Society for Photogrammetry and Remote Sensing.
- [Mor, 1977] Mor, J. (1977). The Levenberg-Marquardt algorithm : implementation and theory. *Lecture notes in mathematics*, 630 :105–116.
- [Morel and Yu, 2009] Morel, J. and Yu, G. (2009). ASIFT : A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2) :438–469.
- [Morsdorf et al., 2004] Morsdorf, F., Meier, E., Kotz, B., Itten, K., Dobbertin, M., and Allgower, B. (2004). Lidar-based geometric reconstruction of boreal type forest stands at single tree level for forest and wildland fire management. *Remote Sensing of Environment*, 92(3) :353–362.
- [Neubert et al., 2007] Neubert, B., Franken, T., and Deussen, O. (2007). Approximate image-based tree-modeling using particle flows. In *ACM SIGGRAPH 2007 papers*, pages 88–es. ACM.
- [Nilsson, 1996] Nilsson, M. (1996). Estimation of tree heights and stand volume using an airborne lidar system. *Remote Sensing of Environment*, 56(1) :1–7.
- [Nistér, 2004] Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6) :756–770.
- [Oh and Kay, 1998] Oh, Y. and Kay, Y. (1998). Condition for precise measurement of soil surface roughness. *IEEE Transactions on Geoscience and Remote Sensing*, 36(2) :691–695.
- [Oram, 2001] Oram, D. (2001). Rectification for any epipolar geometry. In *Proc. British Machine Vision Conf.*, pages 653–662.
- [Petitpas et al., 2010] Petitpas, B., Beaudoin, I., Roux, M., and Rudant, J. (2010). Roughness measurement from multi-stereo reconstruction. *ISPRS Symposium on Photogrammetric Computer Vision and Image Analysis Saint-Mandé, France*, 38 :104–109.

- [Pollefeys et al., 1999] Pollefeys, M., Koch, R., and Gool, L. (1999). Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters. *International Journal of Computer Vision*, 32(1) :7–25.
- [Pollefeys et al.,] Pollefeys, M., Koch, R., Vergauwen, M., and Van Gool, L. Metric 3D surface reconstruction from uncalibrated image sequences. *3D Structure from Multiple Images of Large-Scale Environments*, pages 139–154.
- [Pollefeys and Van Gool, 1997] Pollefeys, M. and Van Gool, L. (1997). Self-calibration from the absolute conic on the plane at infinity. In *Computer Analysis of Images and Patterns*, pages 175–182. Springer.
- [Quan et al., 2006] Quan, L., Tan, P., Zeng, G., Yuan, L., Wang, J., and Kang, S. (2006). Image-based plant modeling. In *ACM SIGGRAPH 2006 Papers*, pages 599–604. ACM.
- [Rabin et al., 2009] Rabin, J., Delon, J., and Gousseau, Y. (2009). A contrario matching of SIFT-like descriptors. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pages 1–4. IEEE.
- [Roose, 1996] Roose, E. (1996). Méthodes de mesure des états de surface du sol, de la rugosité et des autres caractéristiques qui peuvent aider au diagnostic de terrain des risques de ruissellement et d'érosion, en particulier sur les versants cultivés des montagnes. *Bull Réseau Erosion*, 16 :87–97.
- [Roy and Cox, 1998] Roy, S. and Cox, I. (1998). A maximum-flow formulation of the n-camera stereo correspondence problem. In *Computer Vision, 1998. Sixth International Conference on*, pages 492–499. IEEE.
- [Rusinkiewicz and Levoy, 2001] Rusinkiewicz, S. and Levoy, M. (2001). Efficient variants of the icp algorithm. In *3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on*, pages 145–152. IEEE.
- [Saleh and Scarpace, 2000] Saleh, R. and Scarpace, F. (2000). Image scanning resolution and surface accuracy ; experimental results. *International Archives of Photogrammetry and Remote Sensing*, 33(B2 ; PART 2) :482–485.
- [Scharstein and Szeliski, 2002] Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1) :7–42.
- [Shepard, 1968] Shepard, D. (1968). A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524. ACM.
- [Smith et al., 2004] Smith, M., Asal, F., and Priestnall, G. (2004). The use of photogrammetry and lidar for landscape roughness estimation in hydrodynamic studies. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35 :714–719.
- [Snavely et al., 2008] Snavely, N., Seitz, S., and Szeliski, R. (2008). Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2) :189–210.
- [Stewénius et al., 2006] Stewénius, H., Engels, C., and Nistér, D. (2006). Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(4) :284–294.
- [Suk and Chung, 1983] Suk, M. and Chung, S. (1983). A new image segmentation technique based on partition mode test. *Pattern recognition*, 16(5) :469–480.
- [Szeliski and Kang, 1994] Szeliski, R. and Kang, S. (1994). Recovering 3D Shape and Motion from Image Streams Using Nonlinear Least Squares. *Journal of Visual Communication and Image Representation*, 5(1) :10–28.

- [Tan et al., 2007] Tan, P., Zeng, G., Wang, J., Kang, S., and Quan, L. (2007). Image-based tree modeling. In *ACM SIGGRAPH 2007 papers*, pages 87–es. ACM.
- [Tombari et al., 2007] Tombari, F., Mattoccia, S., and Di Stefano, L. (2007). Segmentation-based adaptive support for accurate stereo correspondence. In *Proceedings of the 2nd Pacific Rim conference on Advances in image and video technology*, pages 427–438. Springer-Verlag.
- [Triggs, 1997] Triggs, B. (1997). Autocalibration and the absolute quadric. In *1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997. Proceedings.*, pages 609–614.
- [Triggs et al., 2000] Triggs, B., McLauchlan, P., Hartley, R., and Fitzgibbon, A. (2000). Bundle adjustment—a modern synthesis. *Vision algorithms : theory and practice*, pages 153–177.
- [Ulaby et al., 1979] Ulaby, F., Bradley, G., and Dobson, M. (1979). Microwave backscatter dependence on surface roughness, soil moisture, and soil texture : Part II-vegetation-covered soil. *IEEE Transactions on Geoscience Electronics*, 17(2) :33–40.
- [van Donk and Skidmore, 2003] van Donk, S. and Skidmore, E. (2003). Measurement and simulation of wind erosion, roughness degradation and residue decomposition on an agricultural field. *Earth Surface Processes and Landforms*, 28(11) :1243–1258.
- [Veksler, 2003] Veksler, O. (2003). Fast variable window for stereo correspondence using integral images. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1. IEEE.
- [Verhoest et al., 2008] Verhoest, N., Lievens, H., Wagner, W., Álvarez-Mozos, J., Moran, M., and Mattia, F. (2008). On the soil roughness parameterization problem in soil moisture retrieval of bare surfaces from synthetic aperture radar. *Sensors*, 8(7) :4213–4248.
- [Viola and Jones, 2002] Viola, P. and Jones, M. (2002). Robust real-time object detection. *International Journal of Computer Vision*, 57(2) :137–154.
- [Wang and Zheng, 2008] Wang, Z. and Zheng, Z. (2008). A region based stereo matching algorithm using cooperative optimization. pages 1–8.
- [Xu et al., 2007] Xu, H., Gossett, N., and Chen, B. (2007). Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Transactions on Graphics (TOG)*, 26(4) :19–es.
- [Yoon and Kweon, 2006] Yoon, K. and Kweon, I. (2006). Adaptive support-weight approach for correspondence search. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(4) :650–656.
- [Zobeck and Onstad, 1987] Zobeck, T. and Onstad, C. (1987). Tillage and rainfall effects on random roughness : a review. *Soil and Tillage Research SOTRD* 5, 9(1).
- [Zribi et al., 2000] Zribi, M., Ciarletti, V., Taconet, O., Paille, J., and Boissard, P. (2000). Characterisation of the soil structure and microwave backscattering based on numerical three-dimensional surface representation : analysis with a fractional Brownian model. *Remote sensing of environment*, 72(2) :159–169.