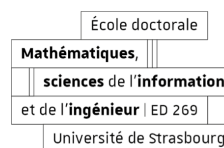




Université de Strasbourg



École doctorale : **Mathématiques, Sciences de l'Information et de l'Ingénieur**

Unité de recherche : **ICube – UMR 7357, INRIA équipe MOSAIC**

THÈSE

Présentée par : **Katia MIRANDE**

Soutenue le : **16 décembre 2022**

Pour obtenir le grade de : **Docteur de l'université de Strasbourg**

Discipline/Spécialité : **Informatique**

Segmentation sémantique et par instance de nuages de points 3D de plantes

Thèse dirigée par :

Christophe Godin

Directeur de recherches, Inria Lyon

Franck Hétroy-Wheeler

Professeur des universités, Université de Strasbourg

Rapporteurs :

David Rousseau

Professeur des universités, Université Angers

Laure Tougne

Professeur des universités, Université Lumière Lyon 2

Examineurs :

Dominique Bechmann

Professeur des universités, Université de Strasbourg

Peter Hanappe

Research leader, Sony CSL

Abstract

Accurate simultaneous semantic and instance segmentation of a plant 3D point cloud is critical for automatic plant phenotyping. Classically, each organ of the plant is detected based on the local geometry of the point cloud, but the consistency of the global structure of the plant is rarely assessed.

In this thesis, we explore the use of two graphs at different scales to segment and classify 3D plant point clouds. At the point scale, the similarity graph allow sus to compute geometrical attributes from the spectrum of the graph and to distinguish between linear organs (main stem, branches, petioles) and two or three-dimensional ones (leaf blades, apices). At the organ scale, the quotient graph is used to obtain a detailed classification and correct potential segmentation defects.

We propose a fast and automatic pipeline that integrates knowledge of the structure of the plant. The method is assessed on both synthetic and real 3D point-cloud data sets of *Chenopodium album* (wild spinach) and *Solanum lycopersicum* (tomato plant).

Keywords : instance segmentation, semantic segmentation, Fiedler vector, quotient graph, phenotyping

Résumé

L'obtention simultanée d'une segmentation sémantique et par instance de nuages de points 3D de plantes est une étape indispensable pour le phénotypage automatique de plantes. Chaque organe de la plante est généralement détecté à partir de la géométrie locale du nuage de points, mais la cohérence globale de la structure de la plante est rarement utilisée.

Dans cette thèse, nous explorons l'utilisation de deux graphes à échelles différentes pour segmenter et classifier des nuages de points de plantes. A l'échelle du point, le graphe de similarité permet de calculer des attributs géométriques basés sur le spectre du graphe et de distinguer les organes linéaires (tige principale, branches, pétioles) des organes à deux dimensions et plus (limbes et apex). A l'échelle des organes, le graphe quotient est utilisé pour obtenir une classification détaillée et corriger des défauts potentiels.

Nous proposons ainsi une chaîne de traitement rapide, automatique et intégrant la structure de la plante. Nous évaluons cette dernière sur des jeux de données de nuages de points 3D de *Chenopodium album* (chénopodes blanc) et de *Solanum lycopersicum* (plants de tomates).

Mots-clés : segmentation par instance, segmentation sémantique, vecteur de Fiedler, graphe quotient, phénotypage

Remerciements

Je souhaiterais remercier toutes les personnes qui m'ont soutenue, même sans en avoir eu conscience, que ce soit par un sourire, un rire ou une discussion.

Je remercie mes directeurs de thèse, Franck et Christophe pour leur supervision et ce sujet de thèse, la segmentation de nuages de points de plantes. Ce sujet m'aura intéressée jusqu'au bout et m'aura amenée à développer de nouvelles compétences et à explorer des domaines palpitants. Je remercie également toutes les personnes brillantes avec qui il m'a été donné de travailler, notamment Fred, Joris, Julie, Guillaume, Romain et Fabrice. Merci à l'équipe IGG de m'avoir accueillie, notamment Dominique. Merci aux membres du jury pour leur participation et pour le temps consacré à mon manuscrit de thèse.

Un grand merci à tous mes collègues doctorants avec qui j'ai partagé un bout de chemin, notamment Nicolas, Pascal, Guillaume, Florian A., Florian I., Paul, Charline. En particulier, Mélinda et Quentin avec qui j'ai partagé un bureau au cours de cette thèse et qui rendaient plus sympathiques même les journées les plus difficiles. Merci également à David de m'avoir épaulée. Je souhaite bon courage à ceux qui n'ont pas encore fini et j'espère qu'ils seront soutenus au quotidien sur la fin comme je l'ai été.

Je tiens à remercier mes parents, qui m'ont mis le pied à l'étrier pour que je ne cesse d'apprendre à réfléchir et à m'exprimer. Je les remercie également pour leur soutien moral. Merci également à ma tante Marie qui a toujours été présente pour moi. Je terminerai par une pensée pour mes grands-parents, qui chacun à leur manière, m'ont beaucoup inspirée.

Table des matières

1	Introduction	1
2	Le phénotypage de plantes	5
2.1	Notre objet d'étude : les plantes	5
2.1.1	Qu'est-ce qu'une plante ?	5
2.1.2	Les organes aériens d'une plante	6
2.1.3	Comment pousse une plante ?	8
2.2	Le phénotypage	10
2.2.1	Qu'est-ce que le phénotypage ?	10
2.2.2	Modéliser des plantes pour mesurer des traits macroscopiques 3D	13
2.3	Le phénotypage dans le projet ROMI	15
2.3.1	Présentation du projet ROMI	15
2.3.2	L'imageur et l'interpréteur de plantes ROMI	16
2.3.3	Plantes à l'étude, données et mesures en sortie	17
2.4	Conclusion	20
3	Segmentation automatique de nuages de points : état de l'art	23
3.1	Généralités sur la segmentation	23
3.2	La segmentation de nuages de points 3D	24
3.2.1	La segmentation par instance de nuages de points 3D	25
3.2.2	La segmentation sémantique de nuages de points 3D	27
3.2.3	En résumé	29
3.3	La segmentation des plantes en leurs différents organes	29
3.3.1	Méthodes basées sur la construction de squelette	30
3.3.2	Les méthodes d'ajustement de modèles géométriques	31
3.3.3	Les méthodes basées sur la génération d'attributs	32
3.3.4	Les méthodes d'apprentissage supervisé et par réseaux neuronaux	32
3.3.5	Cohérence botanique de la segmentation	33
3.4	Conclusion	34
4	Segmentation géométrique basé sur le partitionnement spectral	37
4.1	Introduction au partitionnement spectral	38

4.1.1	La structure de graphe	38
4.1.2	Algorithme de partitionnement spectral classique	40
4.1.3	Interprétation du laplacien et de ses vecteurs propres	41
4.1.4	Temps d'exécution	44
4.1.5	Résultats	46
4.1.6	Utilisation de l'information contenue dans les vecteurs propres	47
4.1.7	Un vecteur particulier : le vecteur de Fiedler	49
4.2	Exploitation du vecteur de Fiedler pour la détection des organes d'une plante	49
4.2.1	Observations sur des graphes simplifiés linéaires	50
4.2.2	Sur des formes géométriques plus complexes	52
4.2.3	Calcul d'un "gradient" du vecteur et observations	52
4.3	Algorithmes de segmentation	55
4.3.1	Application de l'algorithme des K-means sur la norme	55
4.3.2	Algorithmes de croissance de région pour l'obtention d'une seg- mentation par instance	59
4.3.3	Outil de filtrage de la segmentation	61
4.4	Conclusion	65
5	Introduction de connaissances botaniques et changement d'échelle	67
5.1	Le graphe quotient : changement d'échelle dans l'exploitation d'attributs	68
5.1.1	La construction d'un graphe quotient	68
5.1.2	Notes sur les attributs exploités	70
5.2	Les algorithmes sur la structure du graphe quotient	71
5.2.1	Division/fusion de nœuds : l'exploitation des directions	71
5.2.2	Maxima locaux : la différenciation apex/limbe	74
5.2.3	Charge foliaire : la détermination de la tige	76
5.2.4	Plus courts-chemins : l'identification des branches et des pétioles	79
5.2.5	Conformité du graphe quotient : amélioration de la segmen- tation par instance	81
5.3	Conclusion	83
6	Chaîne de traitement et évaluation	87
6.1	Chaîne de traitement	87
6.2	Jeux de données de validation	90
6.2.1	Jeu de données des chénopodes réels	90
6.2.2	Jeu de données des chénopodes virtuels	92
6.2.3	Tomates réelles issues de Phéno4D	93
6.3	Résultats	94
6.3.1	Métriques d'évaluation	94

6.3.2	Evaluation de la segmentation sémantique	96
6.3.3	Évaluation de la segmentation par instance	101
6.3.4	Extraction de traits phénotypiques pour l'exploitation agronomique	102
6.4	Discussion et conclusion	104
7	Conclusion	105
	Bibliographie	111

Introduction

Chaque plante est différente et dispose d'une architecture soit d'une *organisation topologique et géométrique* (Godin, 2000) qui lui est propre. L'architecture des plantes a toujours été d'intérêt dans l'histoire humaine, et notamment pour identifier les plantes pouvant être mangées, les plantes toxiques ou encore les plantes qui peuvent soigner. Néanmoins, dans notre contexte actuel, celui du réchauffement climatique et des bouleversements environnementaux, nous ne pouvons plus nous contenter d'identifier les espèces présentes dans nos forêts ou dans nos champs. Il est nécessaire de prédire quelles sont les plantes qui résisteront dans le futur, à de nouvelles conditions, pour les préserver et encourager leur prolifération (PIAO et al., 2019). Aujourd'hui, les progrès réalisés dans la génétique permettent de déterminer l'ensemble des caractères génétiques d'une plante, son génotype. Cependant, la connaissance du génotype seul, ne permet pas d'anticiper la façon dont va pousser une plante dans un environnement donné. C'est pourquoi, il est nécessaire de déterminer le phénotype d'une plante, soit son architecture et ses caractéristiques physiques et biologiques dans un environnement donné. Ainsi, on peut comprendre quels sont les éléments du génotype qui vont s'exprimer ou non selon l'environnement. Le phénotype d'une plante doit donc pouvoir être quantifié. On parle de traits phénotypiques pour désigner les traits mesurables comme les surfaces foliaires ou la taille de la plante. Pour être pertinentes, ces mesures doivent être effectuées sur un nombre conséquent de plantes. Afin de mesurer ces traits, il est donc nécessaire de disposer d'outils capables d'effectuer de telles mesures, sur un grand nombre de plantes.

Une approche largement adoptée consiste à essayer d'acquérir un modèle, une reproduction informatique de la plante étudiée pour effectuer les mesures sur le modèle de façon automatique et non pas directement sur la vraie plante ; notamment par l'utilisation de systèmes d'acquisition sur les plantes qui s'est démocratisé ces dernières années. Il peut s'agir d'utiliser un smartphone pour identifier une plante rapidement (BERTRAND, 2018) ou d'utiliser des caméras de profondeur pour surveiller la croissance de plantes dans un champ ou en laboratoire (CHÉNÉ et al., 2012). Cette diversité des systèmes d'acquisition a ouvert la porte à la mesure automatique et sans destruction de traits caractéristiques des plantes imagées, autrement dit, au phénotypage automatisé par ordinateur à partir des acquisitions/modèles obtenus. Néanmoins, lorsque les traits phénotypiques à mesurer sont complexes (surface foliaire, longueur de tige, distance entre nœuds, etc.), les organes doivent être préalablement correctement identifiés.

Deux éléments sont nécessaires pour parvenir à mesurer ces traits automatiquement à partir de l'acquisition réalisée. Premièrement, les modèles doivent être décomposés en éléments correspondant aux organes de la plante. Deuxièmement, chaque élément doit être identifié comme l'organe qu'il représente. La première étape est souvent désignée comme de la "segmentation par instance" et la deuxième étape comme de la "segmentation sémantique" ou encore de la "classification".

Lorsque les traits à mesurer relèvent de la structure en 3D de la plante (surfaces foliaires, distance inter nœuds) ; il est préférable de disposer d'une reproduction informatique en trois dimensions soit d'un modèle 3D. Ces systèmes d'acquisitions présentent souvent le même modèle 3D en sortie : un nuage de points. Il est possible de trouver dans la littérature un certain nombre d'approches algorithmiques pour réaliser une segmentation de nuages de points de plantes. Une méthode très présente dans la littérature est l'approche basée sur la réalisation d'un squelette. Elle s'effectue le plus souvent en deux étapes, une étape d'initialisation du squelette, puis une étape d'amélioration de ce dernier. Ces étapes sont basées sur la topologie et des aspects géométriques simples de l'objet (Chaudhury et Godin, 2020). Cette approche présente des difficultés lorsqu'il s'agit de segmenter les feuilles des plantes et surtout de traiter différents types de feuilles. Parmi les méthodes algorithmiques, on note qu'il existe également le partitionnement spectral, une approche non spécifique aux plantes mais qui a fait ses preuves sur cet objet dans la littérature (HÉTROUY-WHEELER et al., 2016), de par sa robustesse au bruit. Notre objectif a été de pousser plus en avant cette approche, en tenant en compte des spécificités géométriques des plantes. L'architecture des plantes est ramifiée avec de nombreux organes linéaires (tiges branches pétioles) et des organes surfaciques (limbes). Enfin, on constate l'émergence des méthodes basées sur l'utilisation de réseaux de neurones. Ces dernières sont basées sur la géométrie et la topologie générale de la forme de la plante. Elles permettent d'obtenir des nuages de points classifiés. La qualité de ces méthodes dépend de la qualité des plantes utilisées dans le jeu de données d'apprentissage.

Les méthodes déjà existantes, qu'elles soient algorithmiques ou à partir de réseaux neuronaux peuvent présenter des défauts de segmentation qui rendent cette dernière botaniquement improbable. Par exemple, une partie du nuage peut être classée en feuille tout en se retrouvant reliée à deux morceaux de pétiole (TURGUT et al., 2022). L'utilisateur de ces méthodes peut s'apercevoir de l'erreur du résultat immédiatement car il a des connaissances sur la plante étudiée et sait quelles sont les caractéristiques de chaque organe de la plante (forme, topologie). On peut constater que la segmentation par instance est très souvent basée soit sur la géométrie locale de la plante, soit sur les caractéristiques de forme des organes (les feuilles sont de

tendance planaires, les tiges sont de forme grossièrement cylindrique). A l'inverse, obtenir une segmentation sémantique nécessite que des classes d'organes aient été définies (par exemple « tige », « limbe »). Des connaissances de la plante étudiée sont donc nécessaires. Ces deux tâches peuvent être résolues de façon indépendante. Néanmoins, il y aurait des avantages à les réaliser de façon simultanée. En effet, les connaissances disponibles sur la structure générale des plantes peut être utile pour corriger ou améliorer la segmentation par instance d'une plante.

Dans cette thèse, notre objectif consiste à coupler la segmentation par instance et sémantique de la plante. Nous souhaitons utiliser des informations géométriques sur la plante et souhaitons intégrer des connaissances botaniques pour identifier les organes et également détecter les erreurs de segmentation par instance. Notre approche utilise deux graphes qui nous permettent d'approcher la segmentation à deux échelles pertinentes. D'une part, le graphe de similarités nous permet de traiter la géométrie locale et d'obtenir un voisinage en chaque point du nuage. D'autre part, nous disposons d'une échelle macroscopique, celle des organes, avec le graphe quotient. Les nombreux points segmentés dans une même instance sont alors représentés en un seul nœud du graphe quotient. Beaucoup de travail a déjà été effectué sur la segmentation par instance et sémantique de nuages de plantes 3D. Néanmoins, les approches proposées sont souvent limitées en terme d'organes identifiés (par exemple uniquement les tiges et les limbes), demandent à l'utilisateur d'intervenir et d'interagir (par exemple : pour créer des jeux de données annotés pour entraîner des réseaux neuronaux) ou ne garantissent pas que la segmentation obtenue est cohérente d'un point de vue botanique (par exemple : un limbe ne devrait pas être directement connecté à la tige principale dans certaines plantes). Nous souhaitons donc proposer une méthode qui permet de segmenter automatiquement les nuages de points 3D en instances classifiées correspondant aux organes de la plante avec une garantie globale que le résultat est botaniquement plausible.

Dans le chapitre 2, nous introduisons les besoins et le vocabulaire propres au phénotypage de plantes. Nous introduisons quelques notions botaniques nécessaires à la compréhension de cette thèse, puis nous introduisons la notion de phénotypage, des raisons qui poussent les biologistes à phénotyper les plantes en grand nombre aux moyens déployés aujourd'hui pour y parvenir. Nous introduisons également le contexte du projet ROMI dont cette thèse fait partie et qui a déterminé les variétés de plantes à l'étude ainsi que la nature des nuages de points 3D sur lesquels nous avons travaillé. Enfin, dans le chapitre 3, nous faisons un état de l'art de la segmentation, plus particulièrement des méthodes de segmentation des nuages de points 3D et des nuages de points 3D de plantes. Nous verrons ainsi que certaines problématiques

sont communes à tous les types de données et quel que soit l'objet d'étude. Néanmoins, certaines méthodes ne sont pas adaptées au cadre de notre étude. Nous nous concentrons sur deux problématiques. La première consiste à trouver des attributs géométriques discriminants en chaque point du nuage. La deuxième consiste à intégrer des connaissances botaniques pour améliorer la segmentation par instance mais également classifier les points. Le chapitre 4 traite de la première problématique. Nous nous intéressons au partitionnement spectral des nuages de points. Nous construisons un graphe de similarité à partir du nuage de points et observons le comportement d'un vecteur propre particulièrement intéressant pour les plantes, le vecteur de Fiedler. A partir de ces observations, nous calculons des attributs nous permettant de différencier les limbes des tiges et les différents éléments linéaires entre eux. Finalement, nous proposons l'utilisation de certains algorithmes non supervisés pour l'exploitation de ces attributs. Néanmoins, on constate que le graphe issu directement du nuage de points présente des limitations pour l'exploitation de certains attributs. Dans le chapitre 5, nous changeons d'échelle d'exploitation du nuage de points et construisons un graphe quotient à partir de la segmentation par instance obtenue dans le chapitre 4. Ce changement d'échelle nous permet également d'introduire des connaissances botaniques, notamment sur l'architecture attendue des plantes. Ainsi, nous identifions les différents organes et proposons une approche pour corriger la segmentation par instance. Enfin, dans le chapitre 6, nous mettons en place une chaîne de traitement grâce aux outils précédemment introduits. Nous introduisons ensuite des jeux de données d'évaluation et nous utilisons ces derniers pour quantifier la qualité des segmentation par instance et sémantique que nous obtenons sur ces trois jeux de données différents comprenant des tomates, des chénopodes réels et des chénopodes virtuels. D'une part, nous évaluons la qualité de la segmentation sémantique sur ces trois jeux de données. Nous évaluons également la qualité de la segmentation par instance sur le jeu de données constitué de chénopodes réels. D'autre part, nous calculons certains traits phénotypiques (surfaces foliaires, hauteur de tige) afin de constater que la méthode nous permet d'obtenir ces traits de façon satisfaisante. Nous concluons alors sur la chaîne de traitement proposée et sur les avantages et limites de cette dernière.

Le phénotypage de plantes

Les plantes sont des organismes vivants qui sont omniprésents autour de nous. Elles sont indispensables à notre survie, qu'il s'agisse de respirer ou de s'alimenter. Dans ce chapitre nous proposons de décrire rapidement ce qu'est une plante et comment elle est organisée. Nous verrons également comment elles sont étudiées à différentes échelles pour améliorer la compréhension que nous en avons mais également pour améliorer les choix qui peuvent être faits en agronomie. Enfin, nous détaillerons dans quel cadre cette thèse évolue et quel est le type de données dont nous disposerons dans la suite.

2.1 Notre objet d'étude : les plantes

2.1.1 Qu'est-ce qu'une plante ?

Le terme plante peut désigner *une espèce végétale dans son ensemble ou encore un végétal de petite taille par opposition à un arbre* (LAROUSSE, 2022). En tant que végétal, il s'agit donc d'un être vivant qui se nourrit principalement voire exclusivement de sels minéraux et de gaz carbonique. On parle de végétaux chlorophylliens ou végétaux verts lorsque le végétal absorbe le gaz carbonique et rejette l'oxygène sous l'action de la lumière, autrement dit, qui réalisent une photosynthèse pour produire leur propre matière organique.

En terme de morphologie, la majorité des plantes possède des organes différenciés. Chaque organe est constitué de tissus, autrement dit d'ensemble organisé de cellules spécialisées. Une plante peut être divisée en deux grands groupes d'organes (GROFF et KAPLAN, 1988) : une partie aérienne et une partie souterraine. Ces deux parties sont en général constituées d'axes organisés en système ramifiés. La partie souterraine est en général la racine. Elle fixe la plante dans le sol et absorbe l'eau et les sels minéraux. Dans la suite de cette thèse nous allons nous intéresser aux parties aériennes de la plante, seules parties visibles.

2.1.2 Les organes aériens d'une plante

La tige

La tige sert de relais, de passage entre les racines et les feuilles, notamment dans l'échange de substances chimiques. On désigne souvent par tige principale, la première tige issue de la germination de la graine (GROFF et KAPLAN, 1988). Lorsque des organes latéraux tels que les rameaux et les feuilles démarrent de la tige, ce point de départ est appelé un nœud (BARTHÉLÉMY et Yves CARAGLIO, 2007). La tige est donc constituée de nœuds et d'entre-nœuds (visibles en figure 2.3B)). Chaque axe d'une plante est construit par un bourgeon qui se situe à l'extrémité de cet axe. Au cours de sa croissance, ce bourgeon va produire des bourgeons latéraux séparés par des segments de tige (entre-nœuds). Ces bourgeons latéraux pourront à leur tour rentrer en croissance et construire des axes latéraux attachés à l'axe original. Ainsi un bourgeon produit en général de proche en proche tout un système ramifié d'axes.

Les feuilles

Une feuille est caractérisée morphologiquement par sa forme aplatie, souvent à symétrie bilatérale. C'est l'organe indispensable permettant la photosynthèse de la plante mais également la transpiration (évaporation de l'eau). Une feuille peut être décrite avec beaucoup de précision et il existe tout un vocabulaire adapté à la caractérisation d'une feuille, notamment détaillé par ELLIS et al., 2009. Nous allons en extraire ici les caractéristiques les plus indispensables pour la compréhension de la suite.

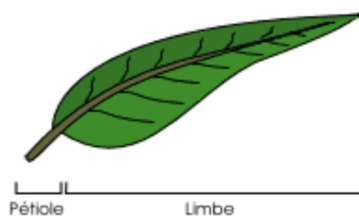


Figure 2.1: Schéma d'une feuille simple pétiolée (Extrait de THEBAULT, 2022)

Il existe trois grands types de feuilles :

- Les feuilles simples pétiolées : comme indiquée en Figure 2.1, cette feuille sera composée d'un limbe (*Partie principale, élargie et étalée, généralement riche en chlorophylle, de la feuille, LAROUSSE, 2022*), d'un pétiole (*Partie de la feuille, généralement rétrécie, qui unit le limbe à la gaine, LAROUSSE, 2022*) et en général,

on observe un bourgeon à sa base (bourgeon axillaire). On peut observer ce type de feuilles sur les groseillers, menthes des champs, basilics sacrés ou encore les chénopodes blancs qui seront utilisés dans cette thèse.

- Les feuilles simples sessiles ou apétiolées : on n’observe aucun pétiole avec ce type de feuille. Le limbe est directement attaché à la tige. C’est le cas du pourpier potager ou du millepertuis.
- Les feuilles composées : elles comprennent des stipules (feuilles réduites à la base du pétiole), d’un pétiole souvent ramifié et de folioles (voir feuilles pennées par exemple sur la Figure 2.2). Si on constate un bourgeon à la base d’un pétiole, il s’agit d’une feuille entière. S’il n’y a pas de bourgeon, le pétiole fait partie d’une foliole. C’est le cas des plants de tomates que l’on verra dans cette thèse ou encore de la reine des prés.

On distingue au sein de ces trois grands types de feuilles des formes de feuilles variées, voir figure 2.2. Les limbes peuvent être très allongées comme les feuilles lancéolées ou falciformes par exemple ou encore être très circulaires comme les feuilles peltées. On note également que si deux feuilles peuvent avoir la même forme générale, il est possible que leur bordure soit différente. Ainsi la géométrie d’une feuille à bordure entière n’est pas la même que celle d’une feuille à bordure épineuse.

Figure 2.2: Quelques différentes formes de feuilles et les différents types de bordures. Extrait de FORTIER, 2022 disponible à l’adresse <https://www.aquaportail.com/definition-3775-feuille.html>.

Les fleurs et fruits

Toutes les plantes ne font pas de fleurs. Parmi les spermatophytes (plantes produisant des graines), c'est le sous-branchement des angiospermes qui est caractérisé par la présence de fleurs qui se transformeront en fruits. On parle d'inflorescence (BELL, 1991) pour désigner un ensemble de fleurs, voisines les unes des autres et à leur mode de groupement.

2.1.3 Comment pousse une plante ?

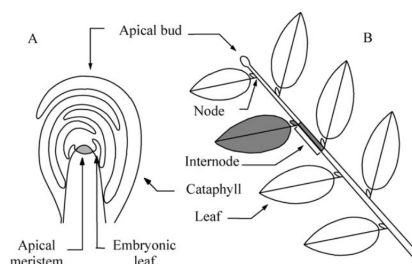


Figure 2.3: Le processus de pousse d'une plante. Extrait de BARTHÉLÉMY et Yves CARAGLIO, 2007. (A) On voit sur ce schéma un méristème protégé par un bourgeon. Des feuilles sont générées. La cataphylle est une feuille dont l'objectif n'est pas la photosynthèse mais est utilisée pour une autre fonction dans la plante. Dans le cas présent, la cataphylle protège le bourgeon. (B) Ce méristème est désigné comme apical car sur la partie la plus haute de la plante. Il contribue donc à sa croissance en hauteur. Les notions de nœuds et entre nœuds sont également illustrées sur la tige de la plante.

Une plante est un organisme dynamique en constante croissance. Au fur et à mesure de la pousse de la plante, on peut voir les feuilles se positionner d'une certaine façon sur la tige ou une branche. Les positions relatives des feuilles sur une plante a un effet direct sur la façon dont elle capte le soleil. La phyllotaxie est le terme utilisé pour désigner cet ordre d'implantation des feuilles (BELL, 1991). Certaines de ces dispositions possibles sont illustrées en Figure 2.4.

La plante s'accroît et prend forme. Cette croissance est marquée également par la perte et la croissance de nouveaux organes (BARTHÉLÉMY et al., 1997). On peut donc voir apparaître des organes temporaires de la plante comme les cotylédons ou des zones de croissances comme les apex. Les cotylédons sont les petites feuilles qui naissent directement de la graine de la plante. Parmi les plantes angiospermes, on peut distinguer deux grands types d'architecture de plantes dès le stade de plantule (début de germination) : les monocotylédones et les dicotylédones. Ces architectures se distinguent par le nombre de cotylédons. Les monocotylédones disposent d'un

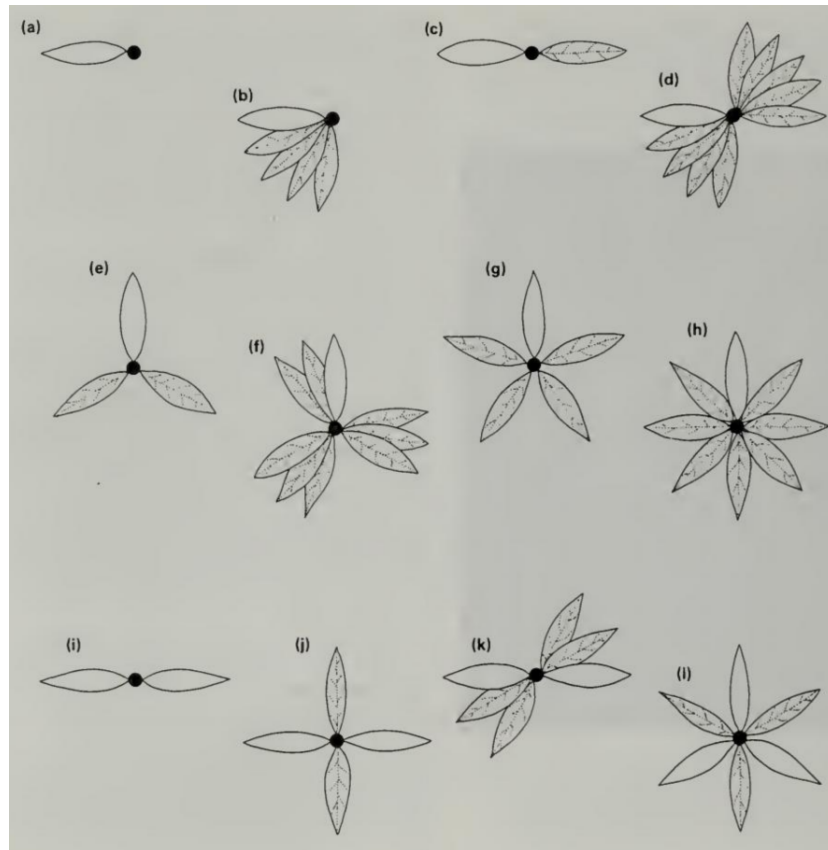


Figure 2.4: Les différentes dispositions des feuilles sur une tige (BELL, 1991). (a) monostique (b) monostique spiralée, (c) distique, (d) distique spiralée, (e) tristique, (f) tristique spiralée, (g) alterne spiralée ou hélicoidale, (h) opposé spiralé ou hélicoidale, (i) opposé, (j) opposé décussé, (k) opposé spiralé, (l) verticillée). Extrait de BELL, 1991.

cotylédon au stade de plantule et ne développent pas de tiges ramifiées mais des feuilles simples apétiolées (par exemples les bananiers, orchidées ou encore les joncs). Les dicotylédones présentent deux cotylédons. Ces cotylédons peuvent être amenés à disparaître ou à rester à la base de la tige de la plante. Dans le cas des chénopodes et des plants de tomates que nous verrons par la suite, ils tendent à rester par exemple.

Si les cotylédons sont essentiels au stade de plantule de la plante, la génération de nouveaux organes est un processus initié par le méristème de la plante. Le méristème est un groupe de cellules indifférenciées (BARTHÉLÉMY et Yves CARAGLIO, 2007, BARLOW, 1989). Chaque axe linéaire de la plante portant des feuilles se termine par un méristème apical. La croissance peut donc se faire en hauteur mais également latéralement. Le méristème est souvent protégé dans un bourgeon. Lorsque l'on parle d'apex chez un végétal, on peut vouloir désigner l'extrémité d'une tige, d'une racine ou encore d'une feuille. Lorsque de nouveaux organes sont générés grâce à un méristème, notamment des feuilles, il se peut qu'ils soient difficiles à différencier les uns des autres initialement. On parle d'apex caulinaire pour désigner cette zone et l'apex le plus

haut dans la plante est parfois désigné par le terme d'apex apical (BARLOW, 1989). De façon générale et pour la suite, on désignera par apex les ensembles de feuilles et entre-noeuds qui n'ont pas terminé leur croissance et sont difficiles à différencier car les pétioles ne se sont pas formées ou que les entrenoeuds ne se sont pas encore allongés.

2.2 Le phénotypage

2.2.1 Qu'est-ce que le phénotypage ?

La sélection de plantes, cultures capables de résister à certaines maladies ou aux changements climatiques en cours est un enjeu crucial pour assurer notre sécurité alimentaire. Aujourd'hui, l'exploitation de la génétique est une opportunité importante dans l'amélioration des cultures (REBETZKE et al., 2019, FURBANK et TESTER, 2011). Afin de comprendre la façon de pousser et l'architecture des plantes, beaucoup de recherches se sont donc portées sur la détermination et la manipulation du génotype des plantes, autrement dit de l'ensemble des caractères génétiques d'une plante. Les technologies traitant de la génomique des plantes se sont fortement développées (L. LI et al., 2014). Cependant, connaître le génotype d'une plante ne suffit pas à comprendre son fonctionnement. En effet, plusieurs individus disposant du même génotype soumis à des conditions environnementales différentes pourront grandir avec des caractéristiques différentes (TARDIEU et al., 2017). Si l'on souhaitait par exemple sélectionner les individus les plus intéressants dans un environnement donné, il faudrait alors étudier son phénotype, autrement dit l'expression visible des gènes de cet individu. Plus précisément, le phénotype est défini comme étant l'ensemble des caractères physiques et biologiques d'un individu. On ne cherche pas simplement à déterminer le phénotype d'un individu donné mais plus généralement son phénotype, soit l'ensemble des phénotypes que l'on peut observer chez cet individu. L'étude de ce phénotype est appelée phénotypage. On comprend que le phénotypage, c'est-à-dire la détermination du phénotype d'un individu, pour être pertinent doit être effectué sur un grand nombre d'individus et donc à haut rendement (FURBANK et TESTER, 2011). Autrement dit, toute mesure qui sera prise doit pouvoir être stockée et à disposition pour étude (SONG et al., 2021).

Lorsque l'on s'intéresse aux plantes, on peut observer ce phénotype à différentes échelles : à l'échelle des cellules, des organes, de la plante entière ou encore de sa canopée (TARDIEU et al., 2017). La figure 2.5 extraite de ZHAO et al., 2019 présente

ces différentes échelles d'observation possibles d'une plante sous la forme d'un schéma. On comprend que de l'échelle d'étude va découler la possibilité d'étudier certains traits phénotypiques (caractères observables du phénotype) que ZHAO et al., 2019 classe ici dans plusieurs catégories différentes. On distingue deux grands types de traits phénotypiques. Les traits internes à la plante vont comprendre les aspect physiologiques de la plante comme les variations d'eau dans les cellules en fonction du temps et des traits structurels comme le taux de croissance à l'échelle cellulaire. Les traits externes à la plante peuvent être morphologiques (volume de la plante, distances inter-nœuds) ou relatifs à la performance (quantité de fruits ou de graines par exemple). On rencontre également les termes micro (HUAN et al., 2021, BRANCH et HAMMONS, 1983) et macro phénotypage (VIJAYARANGAN et al., 2018) pour désigner les traits qui relèvent des cellules de la plante d'une part, et les traits à l'échelle de la plante entière et dans le domaine du visible d'autre part.

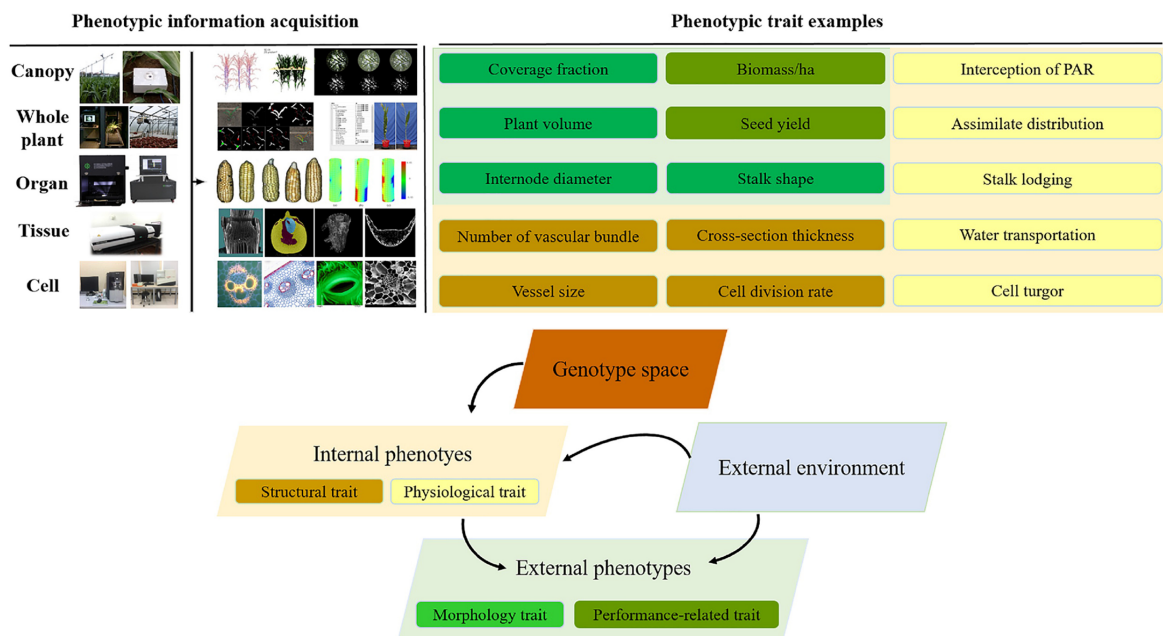


Figure 2.5: Schéma des interactions génotype-phénotype-environnement et des différentes échelles d'observation possibles de la plante (Extrait de ZHAO et al., 2019)

Il est possible que l'on ait besoin de mesurer des traits caractéristiques à différentes échelles de chacune de ces plantes. On comprend qu'il est nécessaire de faire appel à des techniques d'imagerie qui seront adaptées aux traits que l'on souhaite observer. En fonction des traits à l'étude, différents systèmes d'imagerie sont aujourd'hui utilisés et ont été répertoriés dans (L. LI et al., 2014). Les capteurs sont variés et vont de simples appareils photos à des caméras infra-rouges, des détecteurs de CO_2 ou encore des systèmes d'imagerie à résonance magnétique. Il est fréquent que plusieurs traits qui ne nécessitent pas les mêmes techniques aient à être mesurés en même temps. On voit donc l'apparition de plateformes de phénotypage. Ces plateformes

vont embarquer différents types de capteurs et avoir des formes variées en fonction de l'environnement dans lequel elles sont mises en place. La figure 2.6 présente les différentes échelles de travail et les différents environnements d'intérêt dans le phénotypage de plantes. L'étude peut avoir lieu à l'échelle d'une plante dans un laboratoire, à une échelle réduite dans une serre (<10 000 plantes), dans un champ expérimental (> 10 000 plantes) ou dans un champ à grande échelle (PAULUS, 2019). SONG et al., 2021 présente également les outils qui doivent être à disposition pour le traitement des données comme les serveurs et la préparation des plantes (sol, obtention des graines, etc).

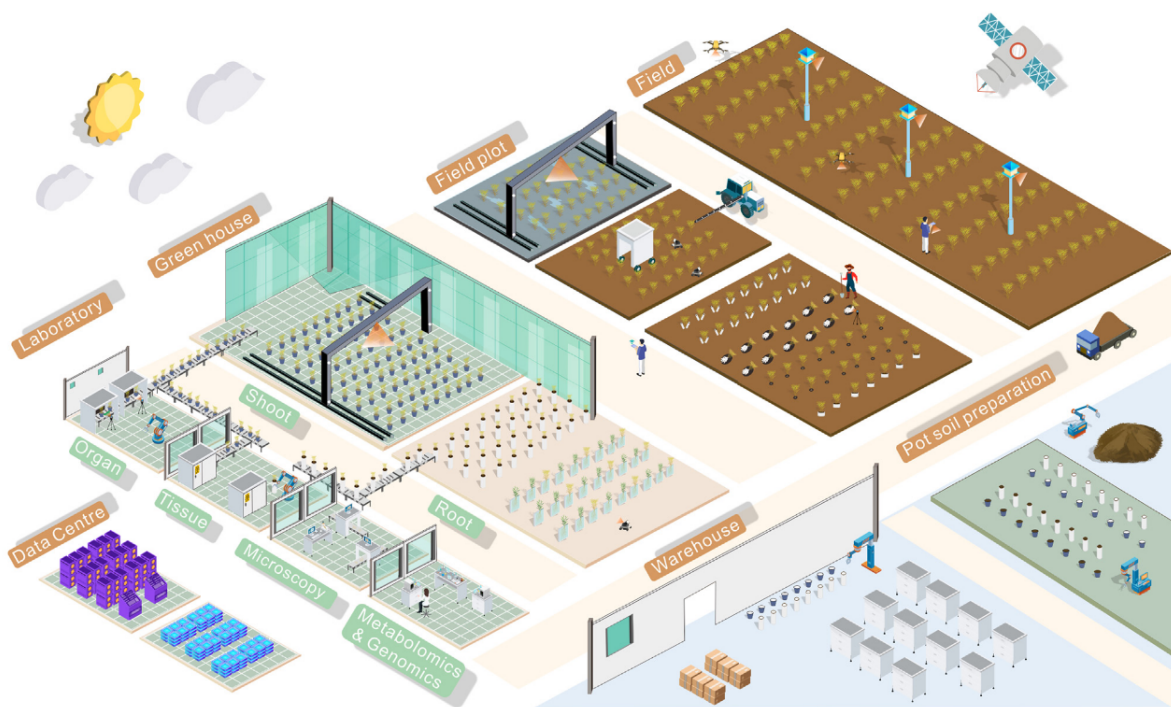


Figure 2.6: Les différents environnements de phénotypage et d'acquisition des données. Extrait de SONG et al., 2021. L'environnement de phénotypage doit s'adapter selon les traits mesurés, les traits microscopiques, à l'échelle des cellules seront observés dans un laboratoire. Les traits macroscopiques peut être mesurés dans une serre, un champ expérimental ou un champ de cultures.

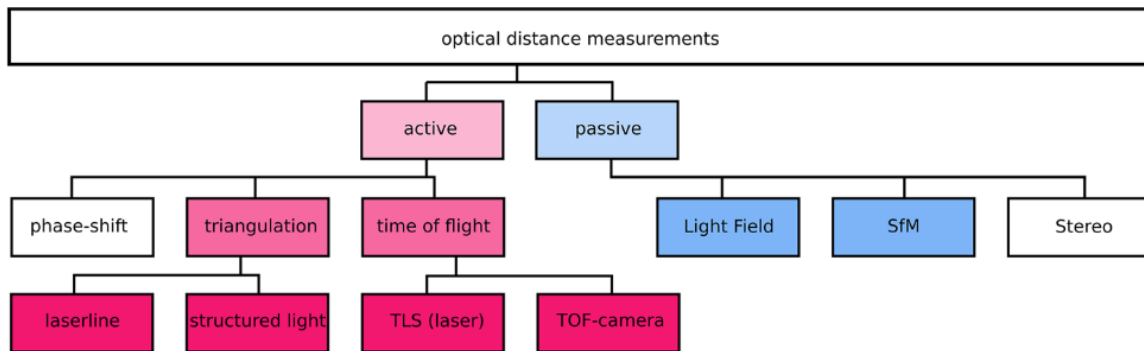
Comparé aux technologies d'imagerie capables de travailler sur la plante dans son entièreté, on constate que ce qui est appelé "micro-phénotypage" (Zhao et al. 2019) à l'échelle des cellules de la plante est difficile à l'heure actuelle et passe très souvent par une étude destructive de la plante, alors que le macro-phénotypage est de plus en plus possible sans destruction de la plante. En macro-phénotypage, il devient donc possible d'observer la même plante à plusieurs temps de croissance différents.

Le phénotypage est donc un sujet vaste auquel on peut encore ajouter les difficultés liées à la variabilité des espèces de plantes notamment dans leurs formes de feuilles ou leur façon de pousser. Un plant de maïs ne pousse pas comme un plant de tomate par exemple, la taille d'un plant de maïs est plus conséquente, ses feuilles sont très allongées et cachent très longtemps l'épi. Les mesures des traits phénotypiques ne peuvent donc pas s'effectuer de la même manière d'une plante à l'autre.

2.2.2 Modéliser des plantes pour mesurer des traits macroscopiques 3D

Parmi tous les traits phénotypiques macro mesurables, on distingue des traits qui relèvent de la forme tridimensionnelle de la plante (surfaces des limbes, longueur des branches ou des pétioles par exemple). Comme mentionné notamment dans CHAUDHURY, WARD et al., 2018, on notera que beaucoup de ces traits géométriques ont été longtemps calculés à partir d'images 2D, particulièrement des photographies. Un grand pan de la littérature est donc dédié à l'exploitation de données 2D en phénotypage (SPALDING et MILLER, 2013, un exemple : AUGUSTIN et al., 2016). Néanmoins, les techniques d'imagerie purement 2D ont des limitations sur certains traits phénotypiques d'intérêt. Par exemple, lorsque l'on calcule la surface d'un limbe à partir d'une image 2D, on ne peut pas considérer la courbure du limbe, c'est-à-dire le fait que ce limbe ne se trouve pas sur un seul plan de l'espace. On obtient donc une surface qui peut être très différente de la surface réelle. Lorsque l'on souhaite mesurer des traits 3D sur des plantes en laboratoire, il est donc préférable de disposer d'un modèle en trois dimensions de la plante pour extraire ce type d'informations. Un modèle 3D peut être obtenu grâce à différentes techniques qui vont dépendre de l'échelle d'étude comme abordé précédemment en figure (2.6).

Les différentes méthodes d'acquisition pour obtenir un modèle 3D de plante sont présentées notamment dans NARVAEZ et al., 2017, PEREZ-SANZ et al., 2017, Z. LI et al., 2020 et PAULUS, 2019. On distingue deux grandes catégories de méthodes d'acquisition selon la classification de PAULUS, 2019 présentée en figure 2.7 en fonction de la source de rayonnement utilisée. Les méthodes dites "passives" regroupent les méthodes qui utilisent uniquement le rayonnement émis par l'objet observé dans son environnement ou le rayonnement solaire réfléchi par l'objet pour l'acquisition. Les méthodes dites "actives" émettent un rayonnement et observent la réaction de l'objet à ce rayonnement.



(a) Classification hiérarchique des méthodes d’acquisition 3D présentant en couleur les méthodes les plus adaptées pour le phénotypage de plantes. Extrait de PAULUS, 2019.

Name	Abbr.	Resolution	Active illumination	Direct point cloud access	Output values	Price
Laser triangulation	LT	< mm	Yes	Yes	XYZ (I)	€€€
Structured light approaches	SL	< mm	Yes	No	XYZI (RGB)	€€
Structure from motion	SfM	mm	No	No	XYZRGB	€
Time of flight	ToF	mm	Yes	Yes	XYZ (I)	€€
Light field measuring	LF	mm	No	No	XYZRGB	€€€
Terrestrial laserscanning	TLS	cm	Yes	Yes	XYZ (I/RGB)	€€€

(b) Les différences entre les méthodes d’acquisitions adaptées au phénotypage de plantes. Les intervalles de prix sont représentés comme suit : € < 1000 Euros, €€ < 10 000 Euros, €€€ > 10 000 Euros. Les valeurs en sortie sont des nuages de points décrits par des coordonnées XYZ, pour lesquelles on obtient parfois une mesure d’intensité (I) ou de couleur (RGB). Extrait de PAULUS, 2019.

Figure 2.7: Bilan des méthodes d’acquisition disponibles pour obtenir un modèle 3D de plante adapté au phénotypage de traits caractéristiques macroscopiques telles que la surface foliaire, les distances entre nœuds ou encore la hauteur de tige.

On constate que toutes ces techniques résultent dans la production d’un nuage de points, soit un jeu de coordonnées en trois dimensions. On note néanmoins que les points du nuage sont parfois augmentés d’une valeur supplémentaire : une intensité (I) ou des couleurs (RGB) (figure 2.7 b)). En effet, tous les modèles en sortie de ces techniques ne se valent pas. PAULUS, 2019 compare et classe ces différentes méthodes d’acquisition permettant aujourd’hui d’obtenir un modèle de plante sur lequel il est possible de mesurer des traits phénotypiques 3D (figure 2.7). Il souligne notamment l’importance de la résolution du modèle pour la mesure de certains traits. Si le modèle n’est pas suffisamment résolu, certains traits comme la surface des limbes par exemple peut être mal évaluée. La précision des traits mesurés est entièrement dépendante de la précision du modèle 3D obtenu. La comparaison des coûts d’acquisition des types de plantes a également été réalisée. Les coûts sont très variables, allant de moins de 1000 euros à plus de 10 000 euros. En résumé, le choix du type d’acquisition dépend de plusieurs facteurs : la quantité de plantes à scanner, le cadre d’acquisition,

la vitesse d'acquisition et la précision nécessaire pour le trait phénotypique à mesurer vont déterminer la résolution à atteindre et le coût d'investissement.

2.3 Le phénotypage dans le projet ROMI

Le phénotypage est un sujet complexe dont les problématiques sont multiples. Le cadre dans lequel notre travail et cette thèse se situe est celui du projet ROMI (RObotics for MIcrofarms).

2.3.1 Présentation du projet ROMI

On observe aujourd'hui en Europe l'augmentation de ce que l'on appelle des micro-fermes. Ces fermes peuvent être trouvées dans des zones rurales mais également péri-urbaines voire des zones complètement urbanisées. Elles ont des surfaces comprises entre 1000 m^2 et 5 hectares et favorisent la variété des cultures, notamment pour éviter l'usage de pesticides. En effet, les productions des micro fermes sont biologiques. Ces micro-fermes sont très productives et viables économiquement (exemple de CARNÉ-CARNAVALET, 2020). Dans ce type de fermes, de par la surface limitée d'exploitation et la diversité des cultures, beaucoup de tâches doivent être effectuées à la main, notamment le désherbage.

Le projet ROMI est un projet de recherche "Research In Action" (RIA) de quatre ans financé par l'union Européenne dans le cadre du plan Horizon 2020 (H2020, 1er novembre 2017 - 31 juillet 2022). Ce projet s'inscrit dans une politique de promotion d'une agriculture locale, durable et à l'échelle humaine. L'objectif est, à court terme, de développer des outils libres et abordables pour l'exploitation des micro-fermes. Deux solutions ont été développés dans ce sens. Le premier est désigné comme le tableau de bord de l'agriculteur. Il est constitué d'une part d'un outil cartographiant l'exploitation et permettant l'identification de chaque plant de culture (le "cable bot"), d'autre part d'un outil permettant une analyse automatique des images obtenues et l'obtention des courbes de croissance c'est-à-dire un outil complet de suivi des cultures. Le deuxième outil a été nommé le rover de désherbage. Il s'agit d'un outil permettant de désherber automatiquement autour des plants de culture en remuant la surface du sol de façon à ce que les mauvaises herbes ne s'y enracinent pas.

A long terme, le projet s'inscrit également dans le développement d'outils de recherche et des moyens de quantifier précisément l'architecture des plantes. Les

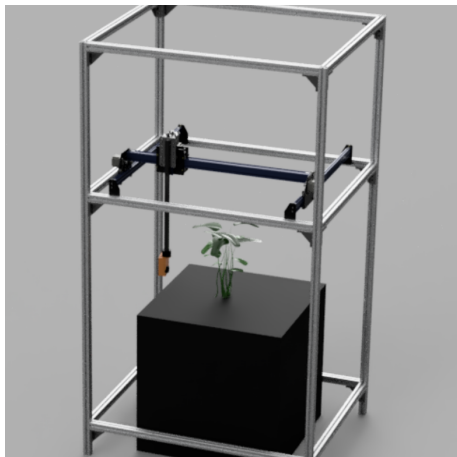
données collectées doivent pouvoir servir à reconnaître les plantes, étudier leur développement, caractériser leur état de santé ou leur âge, etc. L'objectif est notamment de fournir aux chercheurs en biologie un outil automatique permettant de ne pas effectuer toutes les mesures de traits phénotypiques à la main. En effet, pour mesurer la hauteur de la tige ou les distances inter nœuds sur une plante à l'étude, les biologistes doivent mesurer à la main ce qui est très fastidieux quand ce n'est pas destructif (exemples d'études avec protocoles de mesures à la main : RAGNI et al., 2008, PEAUCELLE et al., 2007, BESNARD, 2011). Pour répondre à ce besoin, les outils développés doivent être abordables pour les laboratoires de recherche, donc de coût modéré. Ils doivent également être capables de reproduire la précision des mesures à la main obtenues dans la littérature.

Les outils développés au sein du projet ROMI dans cette optique sont l'imageur de plantes et l'interpréteur de plantes. Le premier outil permet d'acquérir des données à partir des plantes en utilisant un appareil photo monté sur un bras robotisé. Le deuxième outil regroupe tous les outils informatiques de reconstruction et d'analyse de ces images pour aller jusqu'à l'extraction de traits phénotypiques. Un autre outil a également été développé dans le projet ROMI permettant non pas d'imager des plantes réelles mais d'obtenir des nuages de points à partir de plantes virtuelles développées grâce aux L-systèmes et au logiciel L-Py (BOUDON et al., 2012). Le Virtual Plant Imager permet de prendre des images de ces modèles virtuels et d'obtenir un nuage de points à partir de poses de caméra simulées, selon la méthode décrite dans Chaudhury et al., 2020.

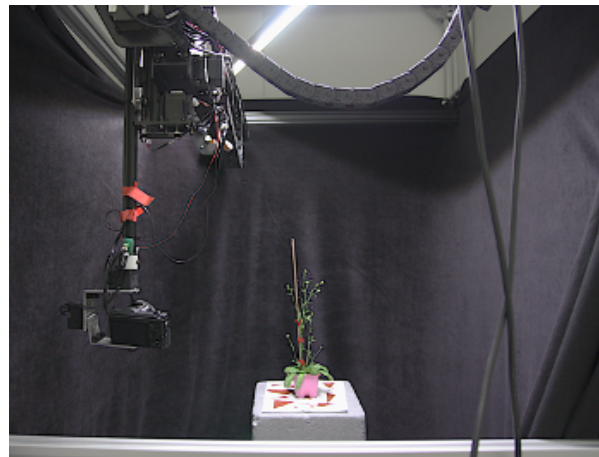
2.3.2 L'imageur et l'interpréteur de plantes ROMI

Étant donné les pré-requis du projet ROMI, la photogrammétrie a été choisie pour créer un nuage de points 3D de la plante qui soit exploitable à un coût raisonnable (voir la figure 2.7 b)).

Ce robot de phénotypage est équipé d'une caméra Sony RX0 RGB (1920 x 1060 pixels). La caméra est fixée à un CNC (Computer Numerical Control) combiné à un bras avec support inclinable. L'ensemble a pour support un cadre en aluminium. L'intérieur du cadre a été tapissé d'un tissu noir pour obtenir un meilleur contraste entre l'arrière plan et la plante modélisée. La scène est éclairée grâce à une lampe LED. Les images sont transmises à un ordinateur disposant du logiciel de traitement via une interface WIFI. L'utilisateur place la plante à étudier dans le scanner et lance le protocole d'acquisition. L'appareil photo se déplace automatiquement autour de l'objet



(a) Modèle 3D de l'imageur ROMI.



(b) Photographie de l'imageur ROMI.

Figure 2.8: Plateforme de l'imageur ROMI. Le scanner est composé d'un cadre en aluminium, d'un CNC combiné à un support inclinable et d'un appareil photo. Extraits de <https://romi-project.eu/>

et prend 72 photographies (protocole non-optimisé pour le moment). Ces images sont chargées dans le jeu de données central.

Les nuages de points 3D sont ensuite reconstruits à partir de cet ensemble de photographies grâce à l'interpréteur ROMI et selon la méthode décrite dans WINTZ et al., 2018. Cette reconstitution passe d'abord par l'estimation des poses de l'appareil photo à partir des images obtenues. Les orientations internes et relatives de l'appareil photo sont effectuées en même temps via un algorithme d'ajustement des faisceaux. Les paramètres de calibration intrinsèques de l'image sont utilisés pour corriger les déformations liées à la forme et au positionnement de la lentille. Les images (toujours en 2D) sont ensuite segmentées. Les parties sombres de l'image qui représentent le drap sombre entourant l'appareil sont détectées dans l'image en appliquant un masque binaire. Les images sont alors en noir et blanc et cela permet de ne récupérer que la plante dans l'image, qui ne sera représentée que par des pixels blancs. Pour reconstruire l'image en 3D à partir des images 2D filtrées la technique de l'enveloppe visuelle de LAURENTINI, 1994 a été utilisée. Le résultat est un modèle constitué de voxels qui est par la suite converti en nuage de points.

2.3.3 Plantes à l'étude, données et mesures en sortie

Les nuages de points issus du pipeline ROMI, de par l'étape de voxelisation sont des nuages dont les points sont disposés sur une grille régulière (voir l'exemple en figure 2.9). De plus, les nuages de points ne disposent pas d'une valeur RGB en

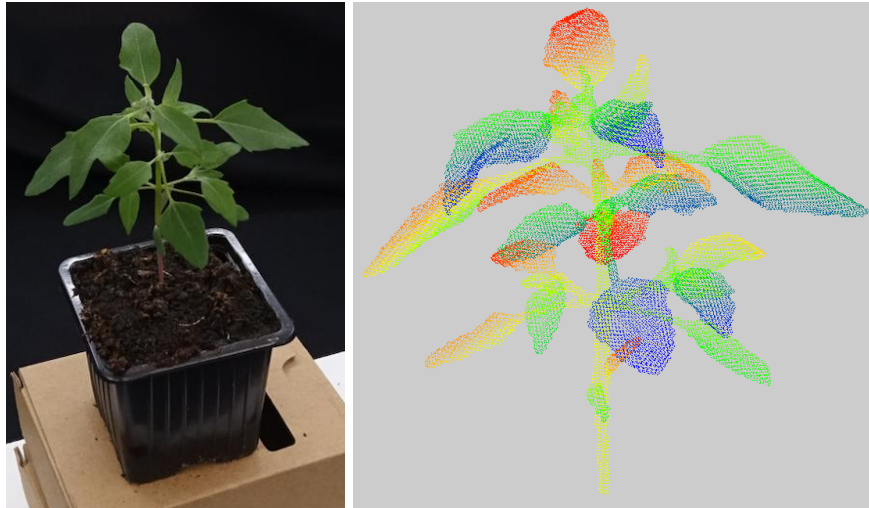


Figure 2.9: Spécimen de chénopode pour tests. La photographie représente la plante dans le robot au moment de l'acquisition. C'est une des images utilisées pour la reconstruction du modèle 3D présenté à droite. Le nuage de points est coloré de façon aléatoire en fonction de l'indexe des points pour obtenir un résultat plus lisible visuellement. Ce nuage est constitué de 55 918 points.

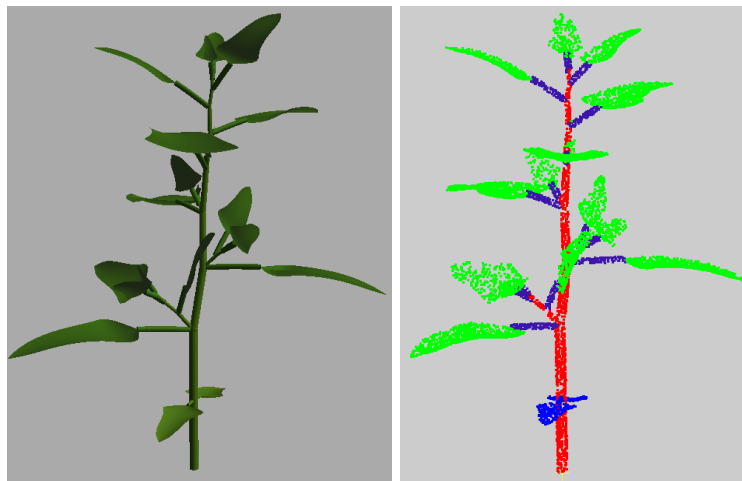


Figure 2.10: Spécimen de chénopode virtuel pour tests. L'image de gauche présente la plante générée dans L-Py. A droite le nuage de points généré à partir de ce modèle. Il est possible de transférer des informations du modèle sur les points du nuage. Les éléments sont donc ici colorés en fonction de trois classes sémantiques, en vert les limbes, en bleu les pétioles et branches, enfin en rouge la tige principale. Ce nuage de points issu d'une plante virtuelle est constitué de 10 000 points.

chaque point car cette valeur n'est pas conservée de par l'utilisation d'un filtre lors de la segmentation des images 2D. La densité de ces nuages est donc constante. La résolution du nuage dépend des paramètres d'entrée de la chaîne de traitement de reconstruction et notamment de la taille des voxels utilisés lors de la dernière étape. Avec l'appareil photo utilisé dans l'imageur ROMI, il est possible d'atteindre une distance entre les points inférieure à 1 *mm*. A l'inverse, les nuages de points issus

de l'imager virtuel ont des points irrégulièrement répartis. La distance entre points dépend entièrement du nombre de points demandé par l'utilisateur en entrée. Nous nous en sommes tenu à des valeurs autour de 10 000 points pour les nuages générés dans ce projet. Un exemple de ce type de nuage est présenté en Figure 2.10.

Deux espèces ont été modélisées dans le projet, que ce soit à partir de plantes réelles ou virtuelles. Il s'agit de l'*Arabidopsis thaliana* et du chénopode. L'*Arabidopsis thaliana* ou arabette des dames (que l'on nommera parfois simplement arabette dans la suite) est une petite plante considérée comme une mauvaise herbe. Elle a un cycle de vie rapide ; il faut 6 semaines à une graine d'arabette pour grandir et produire d'autres graines. Ce cycle de vie rapide combiné à un génome petit fait qu'elle a été très étudiée et est depuis 1998 un organisme modèle de référence en recherche biologique et génétique (WIENKOOP et al., 2010, MEYEROWITZ, 1987). Il s'agit de la première plante dont le génome a été séquencé. Il est donc très intéressant de pouvoir déterminer l'ensemble de son phénotype architectural automatiquement. Un exemple d'*Arabidopsis* et de ses modèles associés est présenté en figure 2.11.



Figure 2.11: A gauche, une photographie d'*Arabidopsis* utilisée pour la reconstruction du modèle 3D présenté à sa droite. Le nuage de points obtenu est à nouveau coloré grâce aux numéro de points pour plus de lisibilité. Ce nuage de points est constitué de 4440 points. Enfin à droite, un nuage de points d'*Arabidopsis* obtenu à partir d'un modèle virtuel est présenté. Il est constitué de 8190 points. On a également pu obtenir une classification sémantique en trois classes, bleue la tige principale, rouge les pétioles, vert les siliques.

Le *Chenopodium album* (ou chénopode) est une plante considérée en occident comme une mauvaise herbe et également hautement indésirable dans les cultures. Elle est néanmoins cultivée d'autres pays du monde pour ses vertus médicinales et alimentaires (POONIA et UPADHAYAY, 2015). En plus de son intérêt agronomique (que ce soit pour le désherbage ou la culture), cette plante présente des caractéristiques et une forme bien différente de l'arabette ce qui présente un intérêt pour s'assurer de la robustesse des algorithmes. Des exemples de chénopodes sont présentés en Figures 2.9 et 2.10.

De façon générale, ces deux plantes sont de petites plantes ramifiées appartenant aux angiospermes dycotylédones. Les nuages de points sont acquis à un stade de croissance relativement jeune de façon à ne constater aucune occultation dans le nuage de points, nous ne traitons pas des problématiques d'organes qui viendraient cacher d'autres organes de la plante. Les fleurs sont donc absentes des nuages. On notera néanmoins que l'on s'intéresse aux plantes lorsque les premières feuilles sont apparues et non pas à leur stade de plantule où seuls les cotylédons de la plante sont visibles. On retrouve sur les deux modèles deux cotylédons à la base du nuage de points. L'ensemble des caractéristiques de développement d'intérêt dans l'étude sont présentés dans le Tableau 2.1.

	Tige	Feuilles	Fleurs	Fruits
<i>Chenopodium album</i>	Anguleuse, simple ou ramifiée	Feuilles inférieures opposées à limbe entier, ovales. Feuilles supérieures sont alternes et proches d'une forme lancéolée.	Petites, verdâtres, groupées en glomérules (inflorescence composée d'un ensemble dense de fleurs sessiles)	Petits fruits ne contenant qu'une seule graine.
<i>Arabidopsis thaliana</i>	Simple ou ramifiée, peu feuillée	Feuilles inférieures sont disposées en rosette ; feuilles obovales, dentées. Feuilles caulinaires sont souvent peu nombreuses, sessiles, elliptiques, lancéolées.	Petites, blanches en grappes.	Siliques grêles et dressés.

Table 2.1: Tableau décrivant certaines des caractéristiques de l'*Arabidopsis thaliana* et du *Chenopodium album* à l'âge adulte. Données synthétisées à partir de INRA, 2000. Les informations sur les organes qui ne nous intéressent pas dans cette étude ont été grisées.

2.4 Conclusion

Les nuages de points qui seront utilisés dans la suite sont issus du projet ROMI et donc de cette chaîne de traitement. Une fois les nuages de points obtenus, l'objectif est de pouvoir mesurer des traits phénotypiques sur ces modèles de plantes. On citera

notamment par exemple dans le cadre de ce projet : les distances inter nœuds, les angles d'embranchement des pétioles/branches avec la tige, les surfaces des limbes, la hauteur de la tige. Certains de ces traits (surface et hauteur de tige) ont été mesurés à l'issue de la segmentation réalisée dans cette thèse dans le chapitre 6. On notera que PAULUS, 2019 distingue deux grandes catégories de traits phénotypiques géométriques de la plante : les complexes et les non complexes. En effet, certains traits comme le volume de la plante ou sa hauteur sont des traits non complexes car ils peuvent s'obtenir directement à partir du nuage de points. Les traits à l'étude dans le projet, à l'inverse, sont qualifiés de complexes, car pour pouvoir effectuer les mesures mentionnées, il faut avoir pu localiser et identifier certains organes de la plante dans le modèle. Il apparaît donc comme nécessaire de pouvoir segmenter et classifier les nuages de points de plantes.

Segmentation automatique de nuages de points : état de l'art

Ce chapitre a pour objectif de comprendre quelles sont les différentes approches possibles pour la segmentation de nuages de points 3D de plantes et quelles sont les difficultés majeures que l'on peut rencontrer dans ce domaine. Pour y parvenir, nous allons tout d'abord aborder la segmentation d'un point de vue général, puis nous nous intéresserons aux méthodes particulièrement utilisées pour les nuages de points. Enfin, nous nous intéresserons à la segmentation de nuages de points de plantes et ce que la connaissance préalable de l'objet d'étude peut apporter à la problématique de segmentation et de classification.

3.1 Généralités sur la segmentation

La segmentation consiste à partager, diviser ou encore couper en segments des objets (LAROUSSE, 2022). De façon générale, la réalisation d'une segmentation de données est exécutée comme indiqué en Figure 3.1. Une généralisation de cette action de segmentation est disponible dans JAIN et al., 1999 et est résumée ici. Les données à segmenter sont composées d'éléments communément désignés comme des vecteurs caractéristiques ou des observations (ou encore des motifs). Un vecteur caractéristique \mathbf{x} est composé de plusieurs attributs x_i tels que $\mathbf{x} = (x_1, x_2, \dots, x_d)$. Dans la première étape présentée en figure 3.1, on cherche d'abord à isoler parmi les attributs disponibles ceux qui sont d'intérêt pour la différenciation des observations et donc pour la segmentation. Il peut également arriver que l'on génère/calcule de nouveaux attributs. Une notion de similarité ou proximité est introduite dans la deuxième étape. Il s'agit le plus fréquemment de la définition d'une distance entre chaque observation qui s'obtient à partir des attributs. L'étape de segmentation peut alors être effectuée avec un très grand nombre d'algorithmes disponibles et s'appuie sur la notion de similarité qui a été définie. En sortie, la segmentation peut être stricte (la donnée est divisée en différents groupes d'observation ("clusters"), ou instances) ou floue (chaque observation peut appartenir avec des degrés de confiance différents à chaque segment).

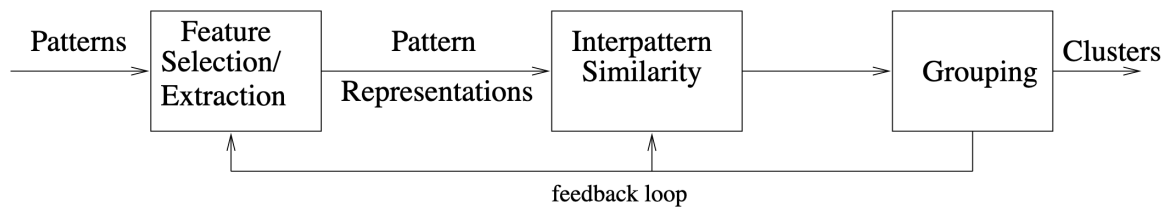


Figure 3.1: Chaîne de traitement générale pour une segmentation de données. Extrait de JAIN et al., 1999.

Une fois les groupes réalisés, il est nécessaire d'évaluer leur qualité. On distingue trois grandes catégories d'évaluations (JANE et DUBES, 1998) :

- L'évaluation externe où les groupes obtenus sont comparés à des groupes de référence, ceux que l'on aurait souhaité obtenir sur les données. Cette évaluation nécessite de disposer d'une segmentation de référence sur ces données.
- L'évaluation interne où les groupes obtenus sont évalués selon des critères intrinsèques aux données. Ces critères sont souvent statistiques. On distingue l'évaluation de la similarité des observations au sein même du groupe (distance intra-groupe) et la dissimilarité, soit la différence entre les observations de différents groupes (distance inter-groupes).
- L'évaluation relative où deux segmentations différentes sont obtenues et comparées sur les mêmes données. Les mérites et défauts de chacune sont caractérisés.

De façon générale, ESTIVILL-CASTRO (2002) montre la diversité des algorithmes existants pour la segmentation dans la littérature et la difficulté que l'on peut avoir à harmoniser les appellations et les méthodes. En effet, la façon dont on approche une segmentation dépend directement de la nature des données que l'on traite et des représentations dont on en dispose. C'est pourquoi nous allons nous concentrer dans la suite sur la littérature autour de la segmentation de l'objet qui nous intéresse, le nuage de points 3D. Nous reprenons le vocabulaire utilisé notamment dans GRILLI et al., 2017, associé à la télédétection, et donc des références qui vont suivre.

3.2 La segmentation de nuages de points 3D

Les nuages de points 3D peuvent se définir comme de simples ensembles discrets de points, autrement dit de listes de triplets de coordonnées. Lorsqu'un tel ensemble de points modélise un objet réel, on peut considérer que le nuage de points est un échantillonnage de la surface de l'objet (BERGER et al., 2017). En fonction de la façon dont a été acquis ce nuage et de la nature de l'objet, le nuage de points peut présenter des défauts voire des caractéristiques différentes (voir section 2.2.2 et

figure 2.7b)). On ajoutera également que le type de défauts du nuage varie selon la méthode d'acquisition et le traitement qui auront été utilisés (BERGER et al., 2017). La segmentation du nuage de points ne sera alors pas abordée de la même façon et c'est ce que nous allons voir dans les algorithmes de segmentation de nuages de points existants.

On distinguera dans cette partie et dans la suite deux types de segmentation. La segmentation par instance consiste à regrouper les observations dans un ensemble de groupes ou d'instances. Dans le cadre des nuages de points de plantes, il s'agit d'associer chaque point du nuage à un élément d'un ensemble $P = \{organ_1, organ_2, \dots, organ_p\}$. Chaque instance est un organe de la plante. La segmentation sémantique, également appelée classification, est définie comme l'action de distribuer par classes, par catégories (LAROUSSE, 2022). On souhaite donc que chaque instance soit nommée. La segmentation sémantique nécessite de connaître l'objet sur lequel on travaille afin de pouvoir définir correctement les classes. Dans le cadre des nuages de points de plantes, il s'agit d'associer chaque point du nuage à un élément de l'ensemble $C = \{limbs, apices, mainstem, petioles, branches\}$.

Nous disposons d'articles d'état de l'art très complets regroupant les différents algorithmes de segmentation de nuages de points. L'article récent de XIE et al. (2020) se veut exhaustif et se base principalement sur les deux références existantes précédemment : GRILLI et al. (2017) et NGUYEN et LE (2013), en y ajoutant des précisions techniques.

3.2.1 La segmentation par instance de nuages de points 3D

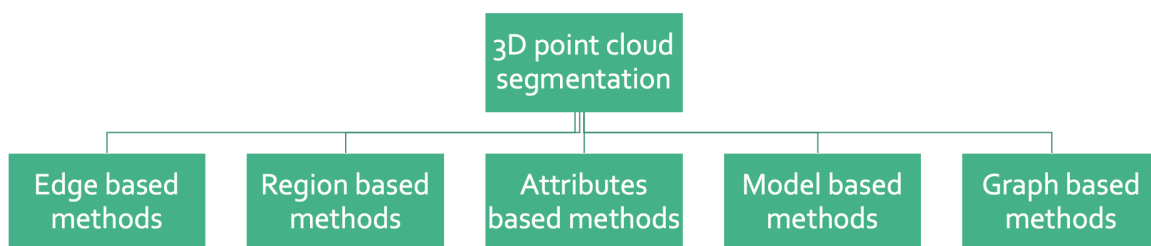


Figure 3.2: Les grandes familles de méthodes de segmentation par instance des nuages de points selon et repris de NGUYEN et LE, 2013.

La figure 3.2 présente une classification possible des méthodes de segmentation par instance utilisées sur les nuages de points. Les algorithmes de segmentation basés sur la détection de frontière entre les points ou encore les regroupements par

croissance de région des points du nuage sont des méthodes largement employées lorsque l'on dispose d'un attribut en chaque point lié à l'acquisition du nuage. Comme nous l'avons vu en section 2.2.2, certaines méthodes d'acquisition comme le lidar mettent à disposition une valeur d'intensité en chaque point (valeur mesurée en fonction de l'intensité du retour du signal lumineux envoyé par le lidar, exemple dans TATOGLU et POCHIRAJU, 2012). Il est également très fréquent de disposer d'une valeur de couleur en chaque point du nuage, ce qui peut également être un attribut en chaque point du nuage permettant de différencier chaque point et donc de devenir un paramètre en entrée de ce type d'algorithmes. Ces algorithmes peuvent également être utiles si le nuage décrit un objet dont les différentes instances sont clairement séparées spatialement. Or, les nuages de points de plantes que nous avons à disposition ne comprennent que les coordonnées en chaque point, sans attributs et sans indices spatiaux pertinents.

On comprend donc qu'il est nécessaire dans ce genre de cas de figure de se tourner vers des méthodes différentes et moins directes. Parmi ces dernières, on peut citer les méthodes d'ajustement de modèle comme la méthode RANSAC. Ces méthodes nécessitent de travailler sur des objets dont la géométrie peut être décrite par des primitives géométriques simples telles que des cylindres, des plans ou encore des sphères. Les méthodes d'ajustement de modèle sont très employées dans la segmentation de nuages de points de rues ou de bâtiments (utilisée par exemple dans le cadre des BIM (Building Information Model) dans MACHER, 2017). Dans le cadre des plantes et plus généralement d'objets organiques, ce principe a rapidement des limites. Les limbes sont rarement descriptibles par un plan car elles sont courbées, la tige principale n'est pas toujours un cylindre bien droit, elle penche souvent et son diamètre évolue. Lorsqu'aucun attribut n'est disponible directement grâce à l'acquisition, il est souvent nécessaire de passer par une méthode de génération d'attributs pour pouvoir obtenir des éléments discriminants entre les différents organes et appliquer une méthode de segmentation.

Contrairement au classement de NGUYEN et LE, 2013, dans le classement de XIE et al., 2020 on notera que les méthodes basées sur la construction d'un graphe ont été regroupées dans une catégorie plus grande, celle des algorithmes de segmentation non supervisée (qui regroupe notamment les K-means que nous verrons plus en détail en section 4.3.1) et qui présentent les mêmes limitations que les algorithmes de croissance de région ou de détermination de frontières. Ils nécessitent de disposer d'attributs discriminants sur les points du nuage. Le classement de XIE et al., 2020 met donc moins en avant les méthodes basées sur la construction d'un graphe à partir du nuage de points alors qu'elles ouvrent à tout un pan de la littérature et de segmentation sur

les objets de graphe. On peut citer par exemple la méthode des Min-Cut utilisée chez GOLOVINSKIY et FUNKHOUSER, 2009. La création d'un nouvel objet graphe permet en plus d'obtenir une notion de voisinage ou topologie entre les points que nous aborderons plus tard en section 4.1.1.

3.2.2 La segmentation sémantique de nuages de points 3D

Lorsque l'on souhaite obtenir une segmentation sémantique (ou classification) d'un nuage de points, on souhaite pouvoir attribuer à chaque point une appartenance à une classe sémantique définie par l'utilisateur. On notera que certaines des méthodes précédentes sont parfois directement suffisantes pour attribuer une classe à chaque point. En effet, si les classes sont distinguables grâce à des attributs dans le nuage dont on dispose ou déjà utilisés pour segmenter les instances, on peut effectuer un classement grâce aux méthodes précédentes. Ces méthodes qui vont identifier les points comme appartenant à un objet particulier du nuage, s'organisent en trois grandes catégories de méthodes selon GRILLI et al., 2017, les méthodes supervisées, les méthodes non supervisées (déjà présentes dans les solutions pour obtenir une classification par instance) et les méthodes interactives.

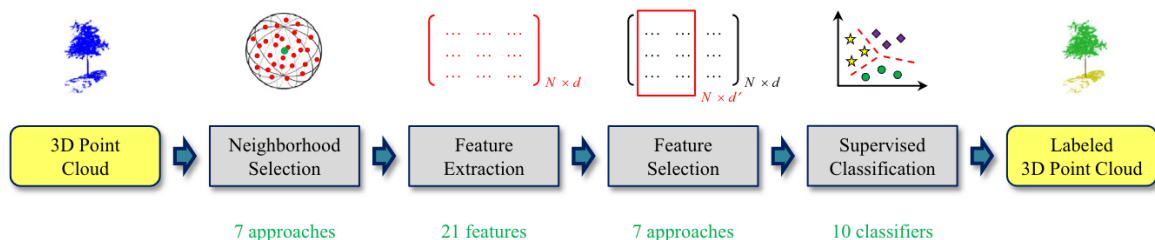


Figure 3.3: Chaîne de traitement pour classifier un nuage de points avec une méthode de classification supervisée extraite de WEINMANN et al., 2015.

Les méthodes d'apprentissage supervisé sur nuages de points sont généralisées dans l'article de WEINMANN et al., 2015. Le schéma résumant comment fonctionne ce type de méthode est présenté en Figure 3.3. Il consiste à calculer des attributs basés sur des voisinages de points tels que la planarité, la linéarité, le "scattering" (WEINMANN et al., 2015, GUINARD et L. LANDRIEU, 2017) mais aussi les normales du nuage par exemple (NURUNNABI et al., 2015). Les normales peuvent être estimées en reconstruisant une surface comme un maillage à partir du nuage de points mais peuvent être également calculées directement à partir du nuage en analysant la matrice de covariance créée à partir des plus proches voisins de chaque point (RUSU, 2009). Des algorithmes de classification supervisée (comme les machines à vecteur de support

dans SECORD et ZAKHOR (2007) ou encore les forêts d'arbres décisionnels dans KIM et SOHN (2011)) sont alors entraînés sur les attributs du nuage que l'on a sélectionné et qui sont le mieux adapté à l'objet de l'étude. Autrement dit, on donne à ces algorithmes un vecteur pour chaque point, comprenant les attributs et la classe dans lequel ce point doit se trouver. Ces algorithmes donnent une règle de classification des points en fonction des attributs auxquels ces derniers sont associés. On observe une volonté de s'affranchir d'un maximum d'interactions avec l'utilisateur dans ces méthodes, au point où l'on constate l'apparition au sein des méthodes supervisées d'une catégorie désignée comme "self-supervised learning" dont un état de l'art est réalisé par Y. WANG et al., 2022 et qui montre le nombre croissant de recherches produites sur ces méthodes.

Si ces méthodes d'apprentissage supervisé connaissent un essor important c'est notamment pour permettre indirectement l'utilisation de réseaux d'apprentissage profond (Y. WANG et al., 2022). Les réseaux neuronaux nécessitent de disposer d'un grand nombre de nuages de points déjà classifiés et la classification à la main de nuages de points est une activité chronophage. Disposer d'une première classification même imparfaite peut donc être une aide précieuse. En effet, l'apprentissage par réseau neuronal consiste pour l'ordinateur à apprendre par l'exemple et en utilisant plusieurs couches qui vont les unes après les autres tenter d'extraire des caractéristiques nécessaires à ce que l'on cherche à avoir en sortie, par exemple ROBERTS et al. (2022). On notera que l'utilisation d'un réseau neuronal avec en donnée d'entrée des nuages de points n'est pas immédiate. Comme expliqué dans QI et al. (2017), ces méthodes sont plus adaptées à des données structurées sur des grilles régulières comme des images 2D ou des éléments 3D en voxels. Les premières utilisations de réseaux neuronaux sur les nuages de points ont donc été réalisées en effectuant des ensembles de voxels. Il faut attendre QI et al. (2017) et le réseau PointNet pour avoir la première méthode prenant en entrée un nuage de points 3D sans conversion en objet ordonné préalable. Aujourd'hui, les méthodes d'apprentissage profond sont largement utilisées sur les nuages de points. Guo et al., 2020 présente un état de l'art de ces réseaux neuronaux et on notera que même si les nuages de points sont utilisés en entrée, ils doivent être réduits à un nombre de points fixe et modéré (quelques milliers). YUAN et al., 2020 présente également un état de l'art pour les données issues de la télédétection qui comprennent les nuages de points mais également les images hyperspectrales par exemple. Ces méthodes nécessitent de disposer de matériel informatique puissant et également de disposer d'un grand jeu de données déjà segmenté pour effectuer l'étape d'apprentissage.

Enfin, il existe également un certain nombre de méthodes qui nécessitent que l'utilisateur intervienne pour valider une classe ou resegmenter une partie. Cette

interactivité est présente dans HÉTROUY-WHEELER et al., 2016 où l'utilisateur dispose d'une interface à la fin de l'algorithme automatique pour fusionner des classes entre elles lorsque ces dernières ont été sur-segmentées.

3.2.3 En résumé

Qu'il s'agisse de segmentation sémantique ou par instance, on note que les algorithmes se basent sur la géométrie ou sur les données d'acquisitions liées au nuage de points. Il est fréquemment nécessaire de générer des attributs discriminants supplémentaires à partir des données ou des attributs existants en fonction de la façon dont on souhaite que les points soient segmentés. Comme souligné par XIE et al., 2020, la génération d'attributs pertinents reste une des grandes difficultés dans la segmentation des nuages de points. D'ailleurs, comme constaté dans la section 3.1, il s'agit d'une problématique qui est générale à la segmentation et pas seulement aux nuages de points. Par ailleurs, une réflexion commune à WEINMANN et al., 2015, GRILLI et al., 2017 et NGUYEN et LE, 2013 consiste à souligner que si les informations déduites du voisinage proche de chaque point sont largement utilisées pour la génération d'attributs, on constate que les points sont plus rarement remis en contexte à une échelle plus grande, celle du nuage entier. Autrement dit, on observe un manque de méthodes consistant à utiliser le contexte spatial des ensembles de points d'un nuage pour parvenir à les segmenter par instance et à les classer. En conclusion, pour mieux considérer l'ensemble du nuage, il faut considérer l'objet qu'il représente et être capables de considérer différentes échelles de traitement, de celle du voisinage immédiat de chaque point à celle de l'objet d'étude entier.

3.3 La segmentation des plantes en leurs différents organes

Même s'il est possible d'obtenir une segmentation par instance uniquement grâce à des éléments géométriques, il n'est pas possible de définir des classes sans connaître l'objet sur lequel on travaille, dans notre cas les plantes. Nous allons ici explorer les différentes méthodes de segmentation qui ont été proposées pour les nuages de points de plantes dans la littérature.

3.3.1 Méthodes basées sur la construction de squelette

La plupart des méthodes appliquées sur les nuages de points de plantes sont largement inspirées des méthodes de traitement des nuages de points. Néanmoins on note qu'il existe aujourd'hui une méthode de segmentation des nuages de points de plantes qui n'est pas présente dans les états de l'art pour les nuages de points en général. Il s'agit de l'approche de segmentation de plantes par squelettisation. Cette grande famille de méthodes est issue des algorithmes d'analyse de formes dont une classification est proposée par LONCARIC, 1998. La construction d'un squelette consiste à définir les formes d'un modèle par un ensemble de courbes centrées dans le modèle (BLUM et NAGEL, 1978, état de l'art général par TAGLIASACCHI et al., 2016). Il existe plusieurs approches pour réaliser un squelette, des approches sont basées sur la construction d'un graphe à partir du nuage de points (H. XU et al., 2007, WU et al., 2019) ou peuvent être basées sur la construction de voxels (utilisée par exemple dans TABB et MEDEIROS, 2018). De façon générale, ces méthodes fonctionnent en deux étapes (comme indiqué en figure 3.4 et dans CHAUDHURY et Christophe GODIN, 2020b), un squelette initial est d'abord construit dans un premier temps puis raffiné dans un second temps.



Figure 3.4: Chaîne de traitement générale pour la reconstitution d'un modèle de plante à partir de nuages de points. Cette chaîne de traitement inclut la construction d'un squelette. A partir de CHAUDHURY et Christophe GODIN, 2020a.

Lorsque le squelette est construit comme en figure 3.5, il permet de récupérer un ordre topologique entre les éléments de la plante (HACKENBERG et al., 2015) et donc parfois d'induire l'appartenance des tronçons à des organes. Néanmoins, la squelettisation n'est pas particulièrement adaptée pour détecter les feuilles puisqu'elle les réduit à un axe linéaire. Comme illustré dans ZIAMTSOV et NAVLAKHA, 2019, les méthodes de squelettisation ne sont pas utilisées seules pour segmenter l'ensemble d'une plante mais plutôt pour la segmentation par instance des parties linéaires de la plante telles que les tiges, les pétioles et les branches.

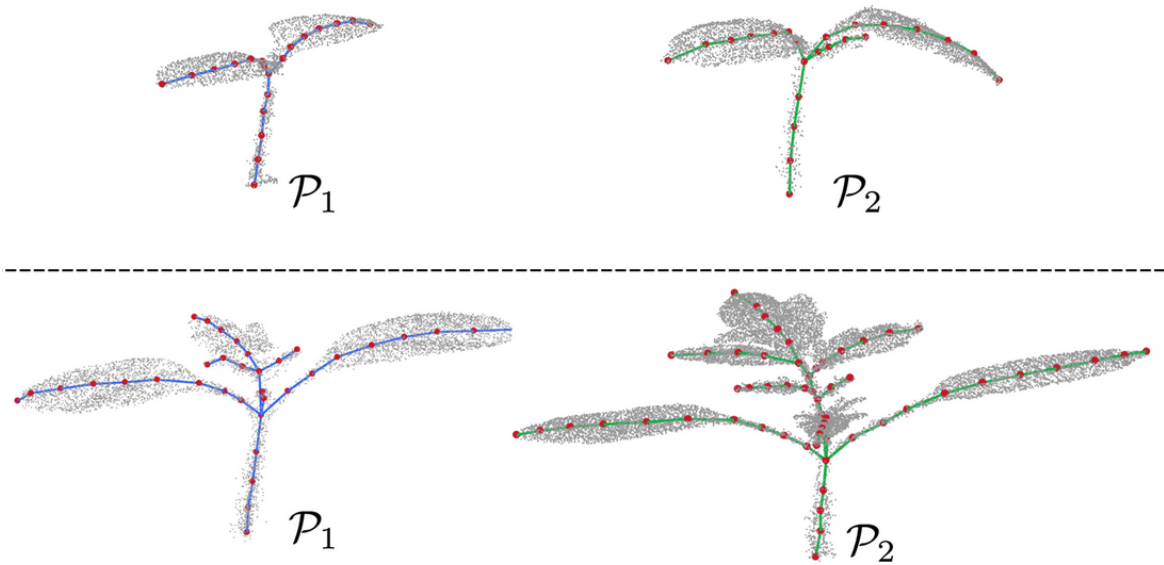


Figure 3.5: Exemple de squelettes superposés aux nuages de points de deux plantes à différents moment de croissance. Extrait de CHEBROLU et al., 2021.

3.3.2 Les méthodes d'ajustement de modèles géométriques

Les premières méthodes de segmentation automatiques et simultanées par instance et sémantique de nuages de points 3D de plantes sont basées sur l'ajustement de primitives géométriques simples pour chaque organe de la plante, par exemple des cylindres pour les organes linéaires tels que les branches et les tiges ou encore les pétioles. On considère les limbes comme des organes plans et on utilise les normales du nuage pour les détecter (ZIAMTSOV et NAVLAKHA, 2019). De façon similaire, GELARD et al., 2017 détecte tout d'abord la tige principale en lui appliquant un cylindre. Les points d'insertion des pétioles et branches sur la tige principale sont ensuite détectés, permettant de segmenter chaque feuille en un pétiole et son limbe. Ainsi que souligné par GHAHREMANI et al., 2021, de telles méthodes manquent de robustesse en présence de bruits ou anomalies d'acquisition. Par conséquent, l'utilisation d'une méthode de traitement basée sur une approche probabiliste a récemment été proposée pour éviter ce problème. La solution alternative à l'ajustement de primitives géométriques notamment proposée par GHAHREMANI et al., 2021 consiste à calculer des primitives géométriques autour de chaque point, en utilisant des points faisant partie du voisinage de chaque élément. On considère non plus qu'un limbe doit être entièrement plan mais qu'il suffit qu'il soit localement plan autour de chaque point.

3.3.3 Les méthodes basées sur la génération d'attributs

A partir du moment où on envisage de calculer en chaque point une valeur illustrant la géométrie locale autour de ce dernier, on rejoint alors le principe des algorithmes basés sur la génération d'attributs. Ces attributs doivent rendre compte de la géométrie du nuage de points de la plante autour de chaque point. Deux grands types d'attributs ont été utilisés dans la littérature. Le premier prend la forme d'un histogramme intitulé le "Point Feature Histogram" (RUSU et al., 2009, PAULUS et al., 2013). La méthode consiste à encoder les propriétés géométriques de chaque point du voisinage du point à l'étude sous la forme d'un histogramme multidimensionnel, chaque point dispose ainsi d'une signature informative. Cette méthode permet de segmenter les nuages de points de plantes en limbes et en tiges/pétiotes (WAHABZADA et al., 2015). Cependant, un travail supplémentaire est requis pour permettre de segmenter chaque organe par instance. Dans VIJAYARANGAN et al., 2018, les deux approches sont utilisées. D'une part des cylindres sont ajustés pour les tiges, d'autre part, un algorithme de croissance de région est utilisé pour distinguer les limbes. Le second type d'attributs repose sur la construction d'un graphe de voisinage construit à partir du nuage de points. Les attributs peuvent alors être calculés de façon locale sur le graphe (ELNASHEF et al., 2019, D. WANG, 2020) ou de façon globale avec les vecteurs propres de la matrice laplacienne du graphe (HÉTROUY-WHEELER et al., 2016, LIU et al., 2020), que nous détaillerons dans le chapitre suivant.

3.3.4 Les méthodes d'apprentissage supervisé et par réseaux neuronaux

Les techniques d'apprentissage par ordinateur peuvent également être utilisées pour obtenir une segmentation simultanée par instance et sémantique de nuage de points de plantes. On observe ces dernières années l'apparition d'un grand nombre de méthodes basées sur les réseaux neuronaux dans le domaine des plantes. Un grand nombre de ces méthodes sont listées dans GUO et al., 2020 dans le cadre de la segmentation de nuages de points en général. Néanmoins, dans le cadre des nuages de points de plantes, ces méthodes ne distinguent en général que deux grandes classes qui regroupent différents organes : les parties linéaires et les parties "feuillues" (ZIAMTSOV et NAVLAKHA, 2019, LIU et al., 2020, D. LI, SHI et al., 2020), la classe "sol" est parfois ajoutée lorsqu'elle n'a pas été enlevée en post-traitement de l'acquisition (SCHUNCK et al., 2021). Cependant, on notera que BOOGAARD et al., 2022 propose une méthode

qui distingue les tiges et branches des pétioles et distinguent également des classes plus complexes : les nœuds, les ovaires, les vrilles et les zones en croissances (que l'on appellera apex ci-après). Cette méthode est basée sur l'architecture PointNet++ (Qi et al., 2017) et développe une stratégie pour contre-balancer le fait que les parties foliaires contiennent bien plus de points que les autres classes. Les segmentations obtenues grâce aux techniques d'apprentissage profond donnent généralement des résultats quantitatifs impressionnants (avec des résultats de classification précis à plus de 95% pour la classe feuille chez BOOGAARD et al., 2022 par exemple). En contrepartie, ces méthodes demandent une quantité importante de données d'apprentissage déjà segmentées (en général, plusieurs centaines de nuages segmentés à la main). Cela représente une tâche manuelle importante et pénible. De plus, l'étape d'apprentissage est chronophage et demande également des machines avec beaucoup de mémoire. Le nuage de points en entrée est souvent ré-échantillonné à quelques centaines de points avant d'être utilisé dans le réseau neuronal qui a besoin de données à taille fixe. Cela introduit une perte de détails et potentiellement une perte des organes les plus petits. De plus, une fois l'apprentissage du réseau neuronal réalisé sur une espèce de plante, nous n'avons pas de garantie que le réseau ainsi entraîné sera efficace sur une autre espèce de plante. Comme on peut le voir sur la figure 3.6, la cohérence botanique des résultats n'est pas assurée en sortie. En (a) et (d) on constate que des points classés en tige sont mélangés et à l'intérieur d'un élément classé en limbe. En (b) une fleur a été faussement détectée entre la tige et les limbes. En (c) on observe des points classés en limbes dans les fleurs. Même dans le cadre des réseaux neuronaux, on constate un besoin d'outil permettant de vérifier la topologie de la segmentation obtenue.

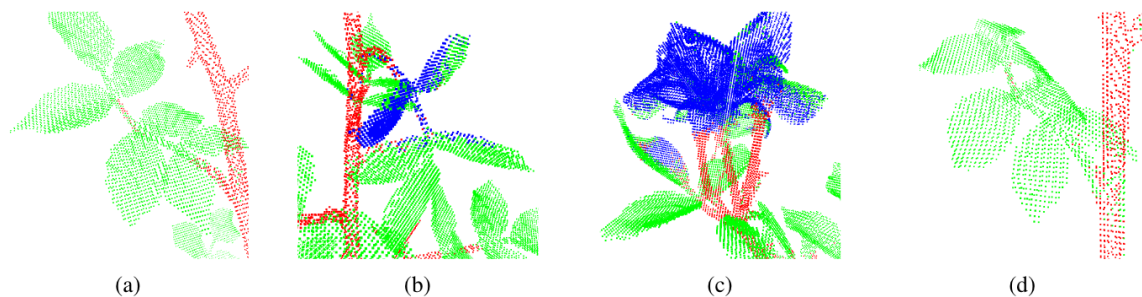


Figure 3.6: Exemple d'erreurs rencontrées avec l'utilisation d'un réseau neuronal (ici PointCNN). Extrait de TURGUT et al., 2022. La segmentation sémantique comprend trois classes : les tiges/pétioles/branches en rouge, les fleurs en bleu et les limbes en vert.

3.3.5 Cohérence botanique de la segmentation

Utiliser des attributs locaux ou des réseaux neuronaux n'assure pas que l'on puisse obtenir une structure globale cohérente de la plante. Pour résoudre cette difficulté

VIJAYARANGAN et al., 2018 détecte les points d'intersection entre la tige principale et les feuilles. HÉTROUY-WHEELER et al., 2016 et LIU et al., 2020 ne vérifient pas si la structure de la plante obtenue est cohérente mais seulement si le nombre d'organes obtenus à la fin du processus de segmentation est suffisant. Dans le cadre de nuages de points d'arbres en forêt, D. WANG et al., 2020 organise les instances en nœuds d'un *superpoint graph*, concept emprunté à Loic LANDRIEU et SIMONOVSKY, 2018. Il optimise ce graphe afin d'identifier les instances qui relèvent du bois de l'arbre. PAPROKI et al., 2012, GELARD et al., 2017 et VIJAYARANGAN et al., 2018 vérifient que chaque feuille est connectée à la tige principale de la plante à la fin de la segmentation. Le nombre de méthodes exploitant la structure de la plante reste faible bien que les articles cités dans cette partie montrent leur intérêt.

3.4 Conclusion

La segmentation de nuages de points 3D par instance et sémantique est une problématique présente depuis longtemps dans la littérature. On note que quelque soit l'objet de segmentation, un des grands enjeux est de déterminer un jeu d'attributs discriminants et cela est vrai pour les nuages de points de plantes également. Le besoin de mise en contexte des points du nuage pour améliorer la segmentation par instance et ne pas uniquement considérer des propriétés locales du nuage est également présent lorsque le nuage représente une plante. En effet, il pourrait être intéressant de pouvoir disposer d'un outil permettant de vérifier la cohérence botanique de la segmentation par instance. Des incohérences de positionnement des organes les uns par rapport aux autres seraient alors détectées et permettraient une correction de la segmentation initiale. Certains travaux ont été réalisés dans ce sens et ont été présentés.

Dans cette thèse nous souhaitons segmenter par instance mais également classifier les points des nuages de points 3D de plantes dont on dispose. Nous ne disposons pas de données d'apprentissage suffisamment grandes pour emprunter l'approche des réseaux neuronaux. Les organes que nous souhaitons détecter (pétioles par exemple) sont souvent de petite taille et ne résisteraient pas forcément à la perte de résolution nécessaire pour adapter le nuage de points à l'entrée d'un réseau. Enfin, cette thèse appartient au projet ROMI présenté en section 2.3.1. Par conséquent, on cherche à développer un outil peu coûteux en moyens et les réseaux neuronaux demandent de faire tourner des ordinateurs sur de longues périodes de temps, donc de l'électricité et des ordinateurs puissants. Nous souhaitons également autant que possible, envisager une méthode adaptable à différentes espèces de plantes. Les méthodes d'ajustement de surfaces présentent des limitations conséquentes sur un certain nombre d'espèces

de plantes de par leur hypothèse forte consistant à considérer les organes des plantes comme suffisamment planes ou suffisamment cylindriques pour être décrites par des formes géométriques simples. Ces méthodes ne sont alors pas adaptées pour les plantes disposant de feuilles plus courbes ou de tiges irrégulières.

Cette thèse a donc deux objectifs : trouver un ou plusieurs attribut(s) discriminant(s) pour obtenir une segmentation par instance du nuage de points mais également introduire des connaissances botaniques pour obtenir une segmentation sémantique du nuage et corriger la segmentation par instance. Notre méthode pour trouver des attributs discriminants se base sur l'utilisation de graphes. Les relations de voisinages entre les points du nuage de points peuvent être exprimées sous la forme d'un graphe (Loïc LANDRIEU et SIMONOVSKY, 2018, WEINMANN et al., 2015. Nous nous appuyons également sur les algorithmes de partitionnement spectral dont l'intérêt pour les plantes a été montré dans HÉTROUY-WHEELER et al., 2016 pour la génération d'attributs pertinents.

La segmentation sémantique peut alors être réalisée, une fois la segmentation par instance obtenue. Les instances sont alors rassemblées pour former des classes, répondant à des critères plus globaux. Les connaissances relevant de la forme globale du nuage de points peuvent être également exprimées sous la forme d'un graphe. L'idée de ces graphes a été introduite par C. GODIN et Y. CARAGLIO (1998). Nous obtenons ainsi un lien entre chaque instance et ses voisins, d'une façon similaire au *superpoint graph* de Loïc LANDRIEU et SIMONOVSKY, 2018 et de D. WANG, 2020. Ce graphe nous permet d'imposer des contraintes botaniques entre les organes lors de la reconstruction de la structure de la plante et permet un contrôle *a posteriori* des règles botaniques dont nous disposons. Dans notre approche, nous nous appuyons donc sur les méthodes présentées qui utilisent les graphes pour segmenter de façon simultanée le nuage de points de plante en instances et étiqueter chaque instance avec une classe sémantique. Cela nous permet d'effectuer plusieurs étapes de correction ou d'amélioration pour récupérer la meilleure structure de plante.

Segmentation géométrique basé sur le partitionnement spectral

Comme nous avons pu le voir dans JANE et DUBES (1998), WEINMANN et al. (2015) ou encore XIE et al. (2020), pour segmenter par instance des données, il est essentiel de disposer d'un attribut pertinent pour obtenir les instances que l'on désire. Autrement dit, dans notre cas, nous cherchons à déterminer un attribut en chaque point du nuage qui puisse nous permettre de regrouper les points décrivant la surface d'un même organe. La génération d'un attribut à partir uniquement des coordonnées 3D du nuage de points ne peut se faire que sur la base de la géométrie de ce dernier car nous ne disposons pas d'autres informations.

Afin d'obtenir des informations de voisinage en chaque point du graphe, nous avons vu que nous pouvions nous appuyer sur la construction d'un graphe à partir du nuage de points (WEINMANN et al., 2015, NGUYEN et LE, 2013). Cette structure de graphe permet alors d'effectuer des méthodes basées sur ce type de structure. HÉTROUY-WHEELER et al. (2016) utilise le spectre du graphe construit à partir d'un nuage de points de plantes afin d'obtenir une segmentation du nuage. Les méthodes de partitionnement spectrales présentent donc un intérêt pour notre objet d'étude et c'est ce que nous approfondissons dans ce chapitre.

Dans une première partie nous verrons, comment construire un graphe de similarité à partir d'un nuage de points puis les grands principes du partitionnement spectral et comment ils s'appliquent aux nuages de points 3D de plantes. Nous verrons les difficultés et limites de ce type d'algorithme et observerons le comportement d'un vecteur propre particulier du spectre du graphe, le vecteur de Fiedler. Enfin à partir de ce vecteur propre, nous verrons comment nous pouvons calculer des attributs d'intérêt pour la segmentation par instance des nuages de points 3D de plantes et proposerons des algorithmes pour les exploiter avec l'aide de la structure de graphe.

4.1 Introduction au partitionnement spectral

4.1.1 La structure de graphe

Notre objet d'entrée est un nuage de points de taille n . On note x_1, \dots, x_n , l'ensemble de points contenus dans le nuage. Nous ne disposons a priori d'aucun attribut permettant de distinguer les points les uns par rapport aux autres, si ce n'est leurs coordonnées spatiales. Nous savons que des points proches en terme de distance euclidienne, soit spatialement, sont des points qui seront le plus souvent dans les mêmes instances d'une segmentation (à l'exception des points aux frontières). Cette distance nous permet d'introduire une notion de similarité $s_{ij} > 0$ entre deux points x_i, x_j . Dans le cadre de notre nuage de points, nous pouvons définir cette notion de similarité avec l'inverse de la distance euclidienne. Ainsi, plus deux points sont proches, plus la similarité est forte.

Il est alors possible de créer un graphe de similarité $G = (V, E)$. Chaque sommet ou nœud $v_i \in V$ représente un point x_i . Nous construisons le graphe non-orienté, c'est-à-dire que les arêtes construites n'ont pas de sens, contrairement aux graphes orientés où les arêtes sont alors représentées par des flèches. Il existe plusieurs solutions pour modéliser un voisinage local entre les données et construire les arêtes E du graphe G , que l'on construit non orienté :

- La définition d'un seuil ϵ de voisinage. Tous les points dont la similarité est supérieure à ce seuil ϵ sont connectés entre eux. Sur notre nuage, cela revient à définir pour chaque sommet v_i une sphère centrée sur v_i et de rayon ϵ dans laquelle tous les points sont considérés comme voisins.
- La définition d'un nombre de voisins K . Chaque point est relié à ses K plus proches voisins. Pour connecter les sommets, on peut considérer que le sommet v_i est parmi les K plus proches voisins du sommet v_j , même si ce n'est pas réciproque. On peut également considérer qu'il est nécessaire que le sommet v_i soit dans les K plus proches voisins de v_j et inversement, pour que les deux sommets soient connectés.
- Enfin, il est possible de connecter tous les sommets entre eux. On pondère alors les arêtes avec les valeurs de similarité s_{ij} . La notion de voisinage locale est disponible uniquement via la pondération et non plus la présence ou l'absence d'arêtes.

Toutes ces solutions sont utilisables pour la suite. Néanmoins, dans le cadre de notre étude, nous avons choisi d'utiliser les K plus proches voisins. Nous choisissons

de considérer qu'un sommet est dans le voisinage d'un deuxième sommet même si ce n'est pas réciproque. On ne construit qu'une seule arête entre deux sommets. Cette méthode nous permet de nous affranchir du calcul de la densité de points du nuage nécessaire à la détermination automatique d'un seuil ϵ dans la première méthode. Nous évitons également de devoir gérer un graphe entièrement connecté comme dans la troisième méthode et cela va nous permettre de manipuler des matrices creuses pour la suite et économiser du temps de calcul. On note les arêtes comme des couples de sommets, ainsi pour tout sommets v_i et v_j leur arête sera notée (v_i, v_j) et par la suite pour alléger les notations on le notera également (i, j) . Les arêtes sont pondérées avec l'inverse de la distance, soit notre notion de similarité. On note la pondération entre un sommet v_i et un sommet v_j , w_{ij} .

D'un point de vue algébrique Il est intéressant de représenter les graphes par des matrices. Ainsi de nombreux calculs sur les graphes peuvent s'exprimer avec des calculs sur des matrices. Pour cela, nous définissons la matrice d'adjacence du graphe. La matrice d'adjacence est une matrice carrée $n \times n$ permettant de représenter un graphe fini. Elle est définie par :

$$W(i, j) = \begin{cases} w_{ij} & \text{si } (v_i, v_j) \in E \\ 0 & \text{sinon} \end{cases} \quad (4.1)$$

On note W la matrice d'adjacence pondérée du graphe. L'élément w_{ij} de cette matrice représente la similarité attachée à l'arête liant les sommets v_i et v_j si elle existe. Par construction, nous n'avons pas d'arête allant d'un sommet à lui même dans E . Par conséquent la matrice W dispose d'une diagonale vide. G est non orienté, la matrice W est donc symétrique.

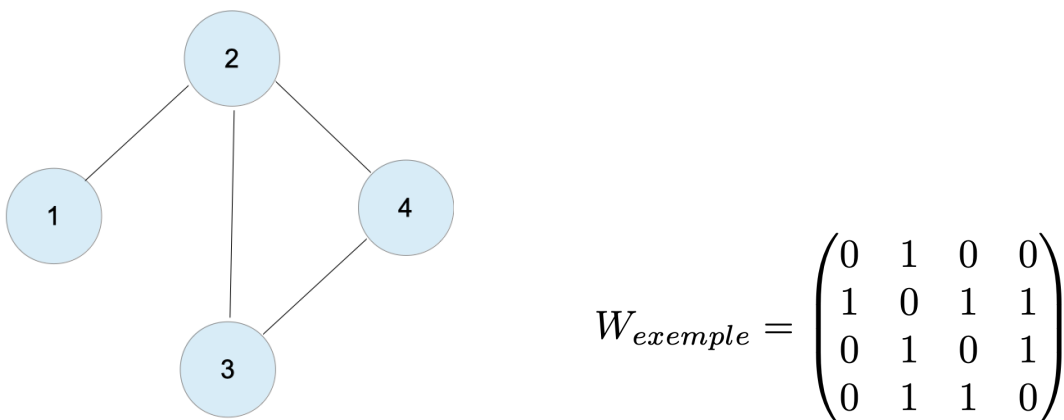


Figure 4.1: Exemple de graphe non orienté et avec une pondération constante égale à 1 avec 4 sommets et sa matrice d'adjacence associée.

Le degré d'un sommet $v_i \in V$ est défini comme la somme des similarités de tous les arcs connectés à ce sommet : $d_i = \sum_{j=1}^n w_{ij}$. L'ensemble des degrés des sommets est représenté dans une matrice diagonale des degrés D . La matrice des degrés du graphe d'exemple en figure 4.1 est égale à :

$$D_{exemple} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

4.1.2 Algorithme de partitionnement spectral classique

L'outil principal utilisé dans le partitionnement spectral est la matrice laplacienne du graphe de similarités. Elle est définie comme suit :

$$L = D - W \tag{4.2}$$

Cette matrice laplacienne contient donc, comme la matrice des degrés, le degré de chaque sommet sur la diagonale. Les termes hors diagonale sont de la forme $-w_{ij}$ si v_i différent de v_j et si v_i et v_j sont reliés par une arête, 0 autrement. Par construction la matrice est de taille $n \times n$. La matrice laplacienne associée au graphe exemple en figure 4.1 est :

$$L_{exemple} = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix}$$

Comme la matrice laplacienne L est symétrique, positive et semi-définie, ses valeurs propres λ_k sont réelles, positives et sont à associées à n vecteurs propres orthogonaux V_k de dimension n , tels que $\forall k = 1, \dots, n$,

$$LV_k = \lambda_k V_k \tag{4.3}$$

Chaque vecteur propre V_k consiste en une liste de valeurs scalaires et chaque valeur de la liste est associée à un nœud du graphe. La taille de chaque vecteur propre est donc de la taille du graphe n (OELLERMANN et SCHWENK, 1991). Ces vecteurs propres ont une interprétation générale sur le graphe : les vecteurs propres correspondants à la valeur propre $\lambda_0 = 0$ permet d'identifier les différents éléments non connectés du

graphe (FIEDLER, 1989). Une seule valeur propre égale à 0 sera obtenue si le graphe est connecté. Le vecteur propre suivant, noté V_F correspondant à la première valeur propre non nulle, sépare le graphe en deux composants correspondants aux nœuds portant des valeurs positives et négatives. Les vecteurs propres suivants continuent à décomposer le graphe G en composants toujours plus petits.

L'algorithme classique de partitionnement spectral consiste alors à utiliser les K vecteurs propres correspondant aux K plus petites valeurs propres et à appliquer un algorithme de partitionnement non supervisé (de type K-means détaillé en section 4.3.1) prenant en entrée le nombre de parties attendues K .

Un algorithme de partitionnement spectral classique peut se décomposer en les étapes suivantes (LUXBURG, 2007) :

- Construction du graphe de similarité. W est sa matrice d'adjacence pondérée.
- Génération de la matrice laplacienne non normalisée L .
- Calcul des k premiers vecteurs propres V_1, \dots, V_k de L .
- Ces vecteurs sont stockés dans une matrice $U \in \mathbb{R}^{n \times k}$ en colonnes.
- Chaque ligne de la matrice U est considérée comme un vecteur $y_i \in \mathbb{R}^k$.
- L'algorithme des K-means est appliqué aux vecteurs $(y_i)_{i=1, \dots, n} \in \mathbb{R}^k$. Les partitionnements P_1, \dots, P_k sont obtenus.

4.1.3 Interprétation du laplacien et de ses vecteurs propres

Une intuition fréquemment utilisée pour comprendre le laplacien d'un graphe et ce que peuvent représenter ses vecteurs propres consiste à comparer la matrice laplacienne à un opérateur laplacien. Ce point de vue est très utilisé dans le cadre des maillages (comme dans SORKINE, 2005).

Pour rappel, l'opérateur laplacien dans un espace continu est défini comme étant la divergence d'un gradient d'une fonction scalaire dans un espace euclidien de dimension n . Dans un système de coordonnées à trois dimensions disposant d'une métrique euclidienne, avec $\vec{i}, \vec{j}, \vec{k}$ des vecteurs unitaires pour les trois axes et x, y, z les coordonnées associées, f une fonction dérivable définie sur tout l'espace et F un champ vectoriel de cet espace, on a :

$$\nabla f = \text{grad} f = \frac{\partial f}{\partial x} \vec{i} + \frac{\partial f}{\partial y} \vec{j} + \frac{\partial f}{\partial z} \vec{k} \quad (4.4)$$

$$\nabla \cdot F = \text{div} F = \frac{\partial F}{\partial x} + \frac{\partial F}{\partial y} + \frac{\partial F}{\partial z} \quad (4.5)$$

$$\Delta f = \text{div}(\vec{\text{grad}} f) = \nabla \cdot \nabla f \quad (4.6)$$

Le gradient permet d'obtenir un champ de vecteur en tout point où la fonction f est différentiable, on mesure ainsi la variabilité de la fonction f au voisinage de chaque point. La divergence est un opérateur qui porte sur un vecteur et dont le résultat est un scalaire. En chaque point, il détermine combien le champ de vecteur, qui peut être considéré par exemple comme représentant le champ de vitesse d'un flot de particules, converge ou diverge.

L'analogie entre l'opérateur laplacien sur une fonction dans un domaine continu et la matrice laplacienne dans le domaine discret a été développée par exemple dans SHUMAN et al., 2013. Afin de pouvoir effectuer cette analogie, une fonction de graphe, ou un signal sur le graphe est défini (figure 4.2). Le signal sur graphe est considéré comme une fonction associant à chaque nœud du graphe un nombre réel. Autrement dit, il est possible d'écrire f comme un vecteur de dimension $(n, 1)$.

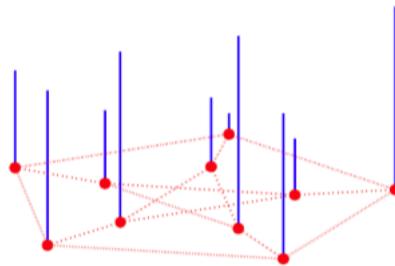
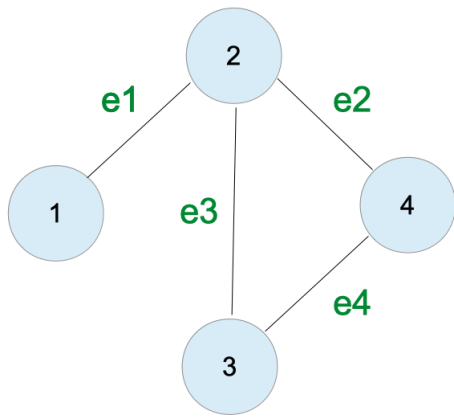


Figure 4.2: Illustration d'un signal de graphe en chaque nœud du graphe de Petersen. La hauteur de chaque barre bleue représente la valeur du signal au nœud sur lequel repose la barre. Extrait de SHUMAN et al., 2013

Dans l'espace, l'opérateur gradient donne une dérivée le long de chaque direction. Dans une structure de graphe, l'équivalent des directions peut être considéré comme étant les arêtes du graphe. Dans le cas discret, la dérivée est calculée comme une différence. Par conséquent, le gradient de la fonction graphe le long d'une arête $e = (u, v)$ peut s'écrire $\text{grad}(f)|_e = f(u) - f(v)$. Il est possible d'obtenir ce gradient le long de chaque arête en introduisant une matrice classique, la matrice d'incidence orientée. Les lignes de cette matrice représentent le nombre de nœuds n et les colonnes le nombre d'arêtes a dans le graphe. Les entrées de cette matrice orientée sont $-1, 0$



$$K_{exemple} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & -1 & -1 \\ 0 & -1 & 0 & 1 \end{pmatrix}$$

Figure 4.4: Une matrice d'incidence orientée correspondant au graphe exemple en figure 4.1

ou $+1$. Une orientation arbitraire est donnée en chaque arête. Comme précédemment, on continue à considérer un graphe dont les pondérations sur les arêtes sont unitaires. Cette matrice permet d'obtenir un gradient tel que $grad(f) = K^T f$, le long de chaque arête du graphe. On obtient un vecteur $(a, 1)$.

La divergence en un point donne la quantité de flux sortant d'un champ de vecteurs. Ce champ de vecteurs est équivalent au gradient de la fonction définie sur un graphe. La divergence d'un champ de vecteur F équivaut à $div(F) = KF$. On obtient alors un vecteur $(n, 1)$ contenant une valeur de divergence pour le champ défini précédemment ($grad(f)$) en chaque nœud du graphe. Les deux opérateurs peuvent être combinés :

$$\Delta f = div(grad(f)) = KK^T f \quad (4.7)$$

$$K_{exemple}K_{exemple}^T = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix} = L_{exemple}$$

L'expression KK^T est une autre façon de définir la matrice laplacienne du graphe, $L = D - W$ (cette expression est notamment utilisée dans MOHAR, 1997). Le produit KK^T est également indépendant de l'orientation choisie pour la matrice d'incidence.

Le calcul des vecteurs propres de la matrice laplacienne conduit à l'obtention d'une base orthogonale qui permet de décomposer le signal f du graphe à la manière d'une transformée de Fourier (notamment dans SHUMAN et al., 2013 qui s'appuie notamment sur des preuves obtenues par HEIN, 2006). On obtient ainsi une décom-

position de la fonction f définie sur les nœuds de G avec les vecteurs propres V_k de la matrice laplacienne du graphe de taille n et $k = 1, \dots, N$.

$$\langle f, V_k \rangle = \sum_{i=1}^n f(i) V_k^T(i) \quad (4.8)$$

Cette intuition est une des raisons pour lesquelles il n'est pas rare de lire dans la littérature les termes de basses fréquences et hautes fréquences ou encore d'harmoniques ou de modes de vibrations (LEVY, 2006) lorsque les vecteurs propres de la matrice laplacienne sont utilisées pour décrire un graphe ou un maillage 3D.

4.1.4 Temps d'exécution

L'objectif de cette partie est d'évaluer le temps de calcul nécessaire pour obtenir la décomposition spectrale d'un graphe construit à partir d'un nuage de points.

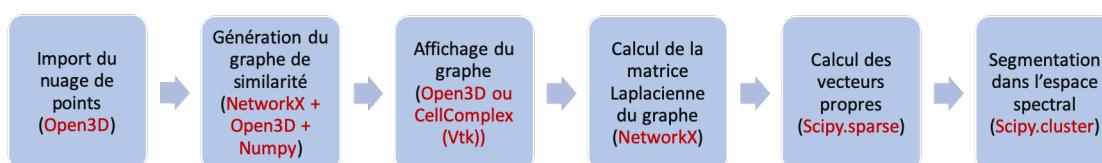


Figure 4.5: Étapes de calcul pour obtenir la décomposition spectrale du nuage de points et l'utiliser pour segmenter le nuage. Les modules Python utilisés pour le calcul sont affichés en rouge.

La figure 4.5 présente les différentes étapes nécessaires pour calculer les vecteurs propres de la matrice laplacienne et segmenter un nuage de points à partir des vecteurs propres de cette dernière. Différents modules Python ont été utilisés, ces derniers ont été choisis de façon à obtenir un résultat le plus rapidement possible.

Nous avons sélectionné 5 nuages de points de tailles différentes et nous avons construit un graphe de similarité à partir de ces nuages en utilisant la méthode des K plus proches voisins décrite précédemment. La figure 4.6 nous permet de constater que le calcul de la matrice laplacienne reste très peu onéreux en temps même sur les nuages de points les plus conséquents. La sélection d'un nombre de voisins plus important augmente le temps de calcul mais ne fait pas passer la barre des 0.01 secondes ce qui est très satisfaisant.

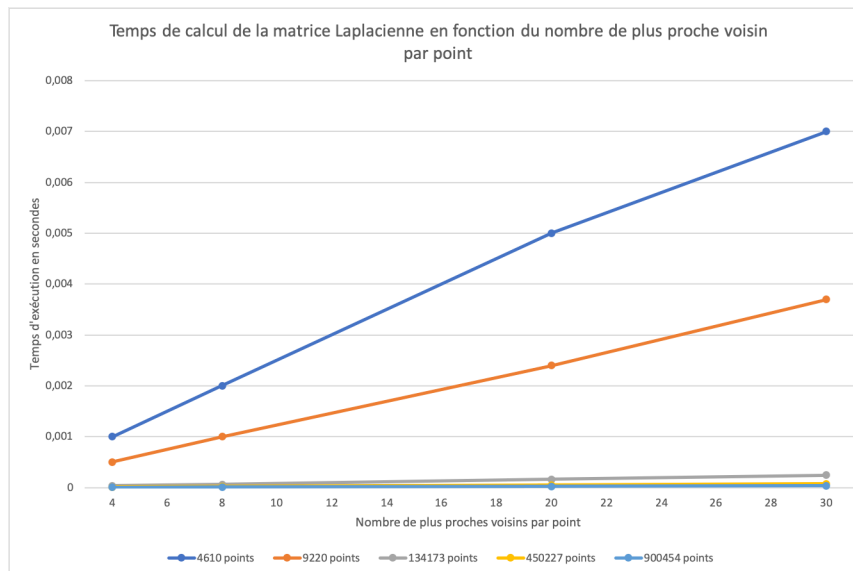


Figure 4.6: Temps de calcul de la matrice laplacienne en fonction du nombre de plus proche voisins par point

La tâche la plus chronophage de cette méthode est le calcul des vecteurs propres. Présenté en figure 4.7, le temps de calcul des vecteurs propres en fonction de la taille de la matrice laplacienne, nous permet de constater que le temps de calcul a une évolution exponentielle en fonction du nombre de vecteurs propres demandés.

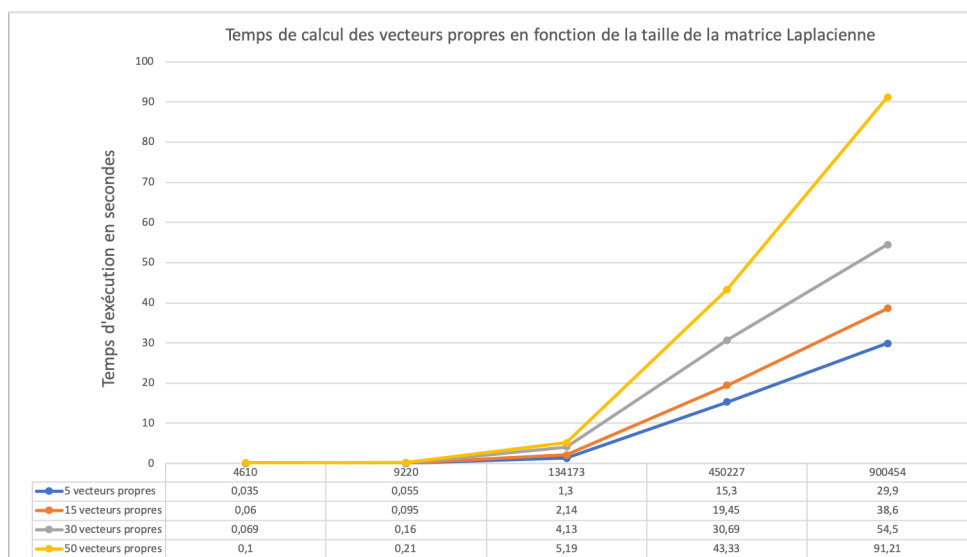


Figure 4.7: Temps de calcul des vecteurs propres en fonction de la taille de la matrice laplacienne, autrement dit du nombre de points contenus dans le nuage.

4.1.5 Résultats

Une fois la décomposition spectrale sur un nuage de points calculée, il est possible de visualiser les vecteurs propres tels que cela a été fait dans la figure 4.8. Les nuages de points ont été colorés en fonction des valeurs de chaque vecteur propre (du bleu au rouge pour les valeurs élevées) et les courbes représentant les valeurs des vecteurs propres en fonction des points ont été tracés. Les premiers vecteurs propres décrivent la forme géométrique du nuage de points de façon très générale. On constate que les vecteurs suivants (exemples des vecteurs 18 et 25 sur la figure 4.8) des points appartenant à des éléments géométriques de détails commencent à être différenciés du reste des points du nuage. Dans le cas, en particulier du nuage de points de l'arabette, ce sont des siliques qui sont distinguables. Les moyens de différencier les différents organes de cette arabette semblent donc bien présents dans la décomposition spectrale du nuage. Les valeurs qui sont associées aux points qui représentent ces organes sont différentes du reste du nuage pour certains vecteurs propres calculés.

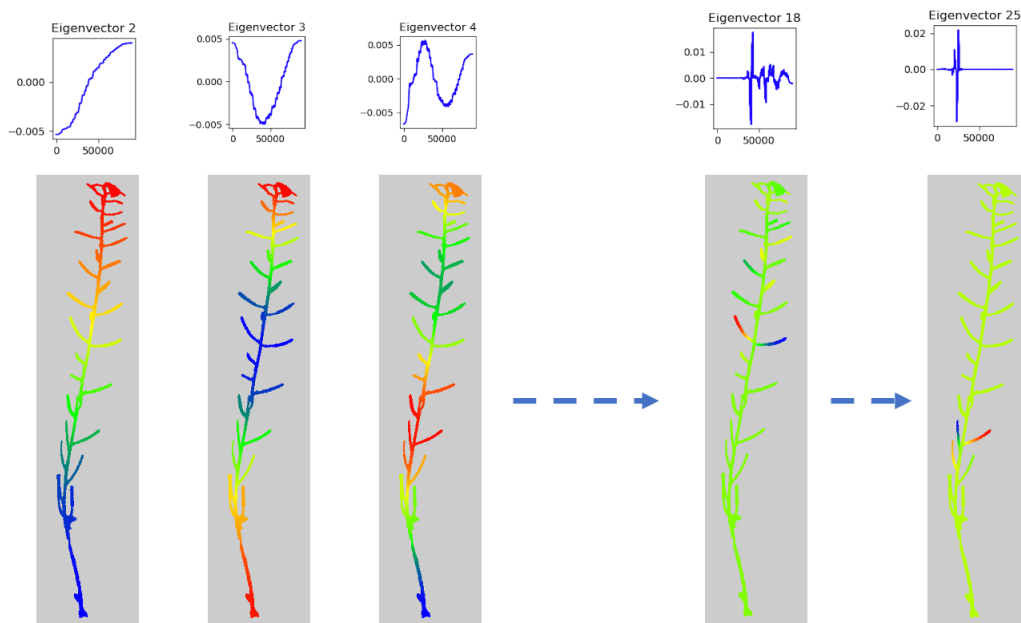


Figure 4.8: Décomposition dans le domaine spectral d'un nuage de point d'arabette type "arête de poisson". Seuls les vecteurs propres 2, 3, 4, 18 et 25 ont été représentés ici. Graphes du haut : les valeurs des composants de chaque vecteur propre représenté en fonction des noeuds (en x). Les noeuds sont triés une première fois pour obtenir une courbe croissante des composants lors de l'affichage du vecteur propre 2, puis cet ordre des noeuds est gardé pour les courbes suivantes. Graphes du bas : Les nuages de points sont colorés en fonction des valeurs de chaque vecteur propre. Les plus petites valeurs sont de couleur bleue et les plus grandes de couleur rouge.

On notera que l'on retrouve dans la décomposition de la figure 4.8 et notamment dans les petits graphiques présentant les vecteurs propres, l'intuition des "fréquences" présentée précédemment. Les vecteurs propres décomposent en parties toujours plus petites le nuage.

4.1.6 Utilisation de l'information contenue dans les vecteurs propres

La question consiste alors à savoir comment extraire les informations qui nous intéressent parmi tous les éléments présentés par les vecteurs propres. L'algorithme classique basé sur la décomposition spectrale et décrit dans la partie 4.1.2 présente une difficulté majeure dans notre cas. Il est nécessaire de savoir combien de parties sont attendues pour appliquer l'algorithme des K-means sur la matrice des vecteurs propres mais également pour connaître le nombre de vecteurs propres à calculer. Les deux éléments, le nombre de vecteurs propres à calculer et le nombre d'instances demandées à l'algorithme des K-means sont présentés comme dépendants et égaux.

Comme le présentent les figures 4.9 et 4.10, le choix d'une part du nombre des vecteurs propres utilisés et d'autre part du nombre d'instances attendues sont cruciaux et impactent de façon importante le résultat de l'algorithme de segmentation.

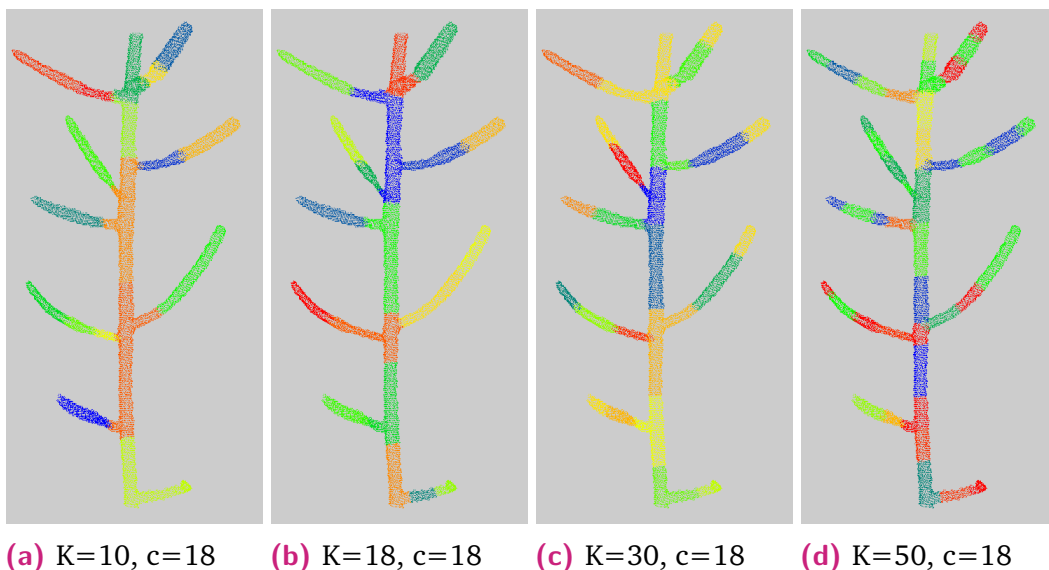


Figure 4.9: Impact du choix du nombre de vecteurs propres sur une segmentation effectuée grâce à la méthode de partitionnement spectrale décrite dans LUXBURG, 2007. K donne le nombre de vecteurs propres utilisés, c le nombre de parties demandées en entrée de l'algorithme des K-means.

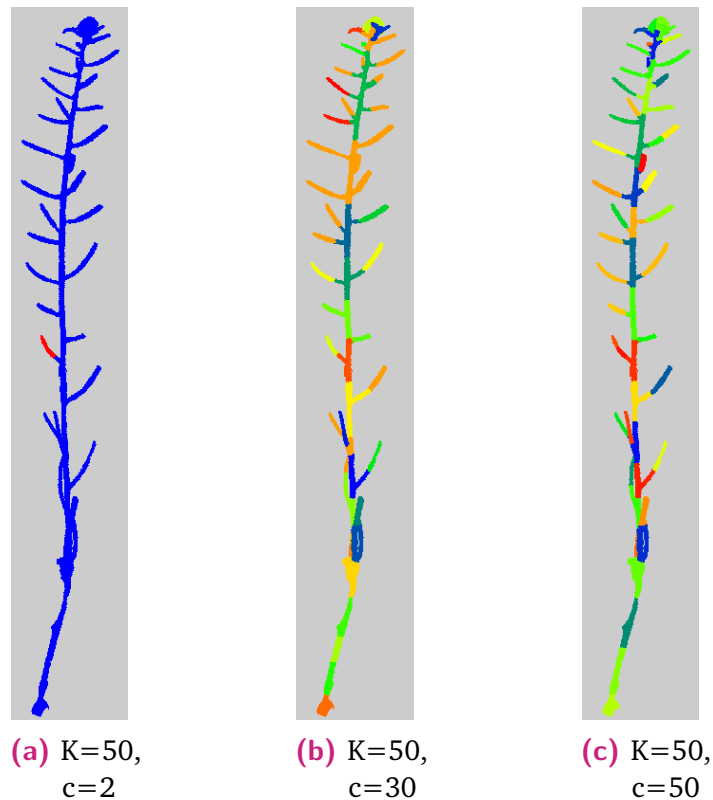


Figure 4.10: Impact du choix du nombre de clusters demandés en entrée de l’algorithme des K-means sur une segmentation effectuée grâce à la méthode de partitionnement spectrale de LUXBURG, 2007. K donne le nombre de vecteurs propres utilisés, c le nombre d’instances demandées en entrée de l’algorithme des K-means.

On notera que dans le cas de figure où l’on donne le bon nombre de parties attendues, c’est-à-dire, le nombre d’organes que nous aimerions voir segmentés dans la plante, on ne parvient pas à obtenir un résultat qui nous satisfasse pleinement d’un point de vue géométrique et biologique. On constate des organes sur-segmentés et d’autres sous-segmentés. Cela peut se constater en figure 4.9b, certains pétioles n’ont pas été séparés de la tige principale, cette dernière est segmentée en plusieurs parties au lieu d’une seule. Les siliques ne sont pas toujours segmentées à l’endroit où visuellement nous pouvons voir la séparation silique/pétiole. Nous n’avons donc que très peu de contrôle sur le choix des informations et détails qui nous intéressent dans la méthode classique du partitionnement spectral décrite dans LUXBURG (2007).

En conclusion, le choix du nombre de vecteurs propres à utiliser selon le niveau de détail géométrique attendu dans le partitionnement spectral classique semble essentiel, autant en terme de résultats, qu’en terme de coût de calcul. Ce problème est considéré comme difficile et est toujours un sujet de recherche dans la littérature (XIANG et GONG, 2008, BLAZÈRE, 2016).

4.1.7 Un vecteur particulier : le vecteur de Fiedler

Pour pallier à la difficulté du choix du nombre de vecteurs propres à exploiter et également de la détermination du nombre d'instances désirées pour pouvoir obtenir une segmentation, nous nous sommes intéressés à un vecteur propre en particulier. En effet, parmi les vecteurs propres composant le spectre du graphe, on constate que le deuxième vecteur propre semble trier les points selon l'axe le plus long du nuage (figure 4.8). Or ce vecteur propre est un vecteur connu dans la littérature et il possède un nom, c'est le vecteur de Fiedler.

Fiedler (FIEDLER, 1973, FIEDLER, 1989) dont le nom a été donné à ce vecteur propre, a étudié les propriétés de la seconde plus petite valeur propre λ_1 et de son vecteur propre correspondant. Il appelle λ_1 la connectivité algébrique du graphe et démontre que cette valeur est plus grande que 0 si et seulement si G est un graphe symétrique non orienté connexe. Il s'agit d'un corollaire au fait que la valeur propre 0 de la matrice laplacienne apparaît autant de fois qu'il y a d'éléments connectés dans le graphe (SLININGER, 2013, ANDREOTTI et al., 2018).

Le vecteur de Fiedler est également présenté comme un excellent outil pour segmenter un graphe en deux (POTHEN et al., 1990). Si l'on reprend le graphe exemple en figure 4.1 des paragraphes précédents et que l'on calcule ses vecteurs propres et valeurs propres à partir de la matrice laplacienne 4.2, on obtient une connectivité de 1. Cette valeur est supérieure à 0, le graphe est bien connecté. Le vecteur de Fiedler associé est $[-8,2 \cdot 10^{-1}; 1,5 \cdot 10^{-16}; 4,08 \cdot 10^{-1}; 4,08 \cdot 10^{-1}]$. Les valeurs négatives sont associées au sommet 1, très peu connecté au reste du graphe. Les valeurs positives sont associées aux autres sommets. Il est donc possible avec les signes des valeurs du vecteur de Fiedler en chaque point de partitionner le graphe en deux. Néanmoins, on notera que le sommet 2, a une valeur très proche de 0, par conséquent, il pourrait être placé dans sa propre classe ou rejoindre la classe du sommet 1 si l'on souhaitait équilibrer les instances (analyse présente dans SLININGER, 2013).

4.2 Exploitation du vecteur de Fiedler pour la détection des organes d'une plante

Afin de pallier aux difficultés présentées sur les algorithmes classiques de partitionnement spectral, notre attention s'est portée sur le vecteur de Fiedler. Ce vecteur

est déjà présenté dans la littérature comment un outil permettant de séparer en deux un graphe. Nous avons pu observer qu'il semblait classer les points du graphe le long de l'axe le plus long du graphe de similarité. Dans les prochains paragraphes, nous allons chercher à caractériser le comportement de ce vecteur et voir en quoi il est exploitable dans la différenciation des différents organes de la plantes.

4.2.1 Observations sur des graphes simplifiés linéaires

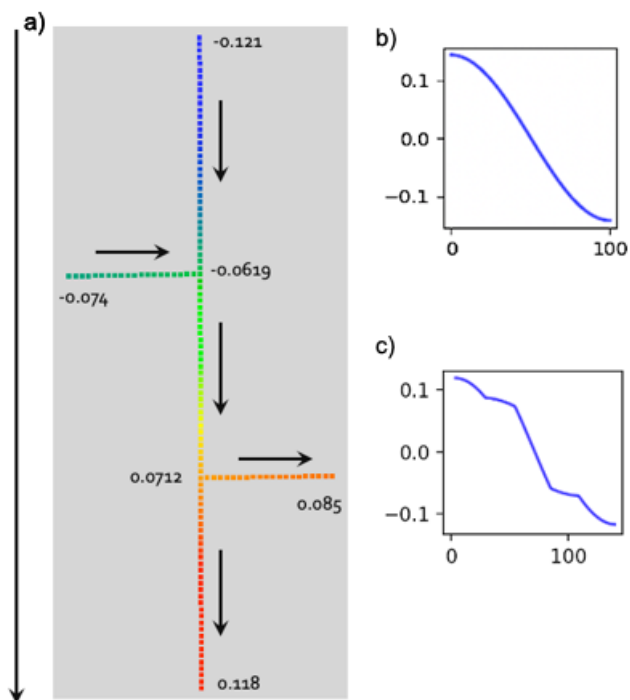


Figure 4.11: Comportement du vecteur de Fiedler sur un exemple de graphe de la forme d'une chaîne élémentaire simple. a) Les différents nœuds du graphe sont représentés par des points colorés. Les arêtes du graphe ne sont pas affichées sur cette illustration. Néanmoins, chaque nœud est relié par une arête aux deux nœuds les plus proches. L'axe principal du graphe est composé de 100 nœuds. Les deux axes secondaires sont chacun composés de 20 nœuds. A chaque nœud a été attribué une couleur en fonction de la valeur que prend le vecteur de Fiedler en ce point. Les valeurs du vecteur de Fiedler aux nœuds aux extrémités du graphe ont été affichés. Le sens de variation positive des valeurs est indiqué par des flèches. b) Ce graphe représente les valeurs du vecteur de Fiedler en fonction des nœuds classés de haut en bas du graphe représenté en a) sans les axes secondaires. c) Le vecteur de Fiedler a été calculé sur le graphe a) entier, avec les axes secondaires.

Nous avons d'abord observé le vecteur de Fiedler sur un type de graphe simplifié, les graphes linéaires simples avec peu de ramifications. Ces chaînes élémentaires sont construites comme un enchainement de nœuds chacun lié à deux autres nœuds les

plus proches. Les nœuds terminaux sont une exception puisqu'ils ne sont liés qu'à un seul nœud. La pondération sur les arêtes du graphe est constante et égale à 1.

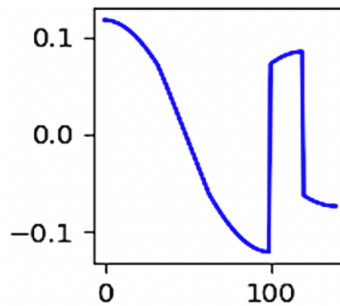


Figure 4.12: Valeurs du vecteur de Fiedler en fonction des nœuds du graphe présenté en figure 4.11 a). Les nœuds des axes latéraux ont été isolés et leurs valeurs affichées à la fin de la courbe, à partir de l'index 100.

Lorsque l'on observe la figure 4.11, on constate que l'ajout de nœuds latéraux à l'axe principal du graphe change la forme de la courbe représentant le vecteur de Fiedler en fonction des nœuds. Les valeurs des nœuds composant les axes latéraux se sont intercalées dans la courbe. Ici, nous connaissons par construction à quels nœuds sont associés les ruptures de pentes que l'on constate en figure 4.11 c). On constate également que l'on atteint un extremum relatif sur chaque axe (minimum ou maximum) à chaque fin d'axe.

L'ordre des nœuds en abscisse étant arbitraire, il nous est même possible de les isoler, ce que nous avons fait en figure 4.12. On constate alors que nous retrouvons la figure 4.11 b). Les valeurs que nous avons isolées correspondent aux points des axes latéraux. Ces ruptures de pentes sont donc induites par la présence des axes secondaires. Nous avons pu constater le même comportement sur des graphes plus longs (100, 500, 1000 nœuds) et en variant la taille et l'emplacement des axes. L'ajout d'axes d'ordre 2 a également été expérimenté. Nous avons pu constater que les pentes sont de plus en plus faibles au fur et à mesure que l'ordre de l'axe augmente.

D'autres propriétés sont également observables sur les figures présentées. On observe notamment une monotonie des valeurs du vecteur le long de l'axe le plus long (en terme de nombre de nœuds) du graphe. Les axes latéraux sont croissants ou décroissants en partant de l'axe principal selon leur position dans le graphe. S'ils sont dans la partie du graphe portant des valeurs négatives, l'axe latéral portera des valeurs décroissantes de l'axe principal vers son extrémité. Si ces axes sont dans la partie du graphe portant des valeurs positives du vecteur de Fiedler, les valeurs sur

les nœuds de ces axes latéraux seront croissantes, de l'axe principal vers l'extrémité. Cela correspond aux principes de segmentation en deux parties d'un graphe grâce au vecteur de Fiedler présenté en section 4.1.7.

4.2.2 Sur des formes géométriques plus complexes

Nous observons le même type de caractéristiques sur des cylindres et sur des plantes en figure 4.13. Le nuage de points en figure 4.13 a) a été construit en deux temps. Une construction de cylindres a d'abord été effectuée sous Blender (COMMUNITY, 2018), puis la surface du modèle a été échantillonnée sous CloudCompare (GPL SOFTWARE, 2022). Le nuage en figure 4.13b est issu de la chaîne de traitement ROMI, il s'agit d'un chéropode réel. Un graphe de similarité a été construit à partir de ces deux nuages de points. Le vecteur de Fiedler présente sur ces deux structures des caractéristiques similaires que ce qui a été observé sur les graphes linéaires simples.

4.2.3 Calcul d'un "gradient" du vecteur et observations

Afin d'observer ces ruptures de pentes et les évolutions de valeurs du vecteur de Fiedler, nous calculons un gradient directionnel moyen (GDM) à partir du vecteur de Fiedler. Soit i un nœud du graphe, x_i les coordonnées de ce nœud et $N(i)$ l'ensemble de ses nœuds voisins, $f(i)$ est la valeur du vecteur de Fiedler au nœud i . Le gradient directionnel moyen du vecteur de Fiedler est alors calculé comme suit :

$$\bar{\nabla} f(i) = \sum_{j \in N(i)} (f(j) - f(i)) \frac{\mathbf{x}_j - \mathbf{x}_i}{\|\mathbf{x}_j - \mathbf{x}_i\|} \quad (4.9)$$

Cette formule définit un champ de vecteurs sur le graphe de similarité que l'on appellera par la suite le GDM du vecteur de Fiedler pour des raisons de simplicité. On peut noter que nous n'avons pas normalisé la formule par le degré du nœud i , néanmoins, tous les nœuds de notre graphe ont à peu de choses près le même nombre de voisins par construction (18). Cette formule nous permet de gérer le cas de figure dans lequel le nuage de points n'est pas uniforme puisque l'apport de chaque point du voisinage dépend de sa position par rapport au nœud i .

Ce champ de vecteur fournit deux attributs puisque l'on obtient une direction et une norme pour chaque vecteur en chaque point du nuage de points et en chaque

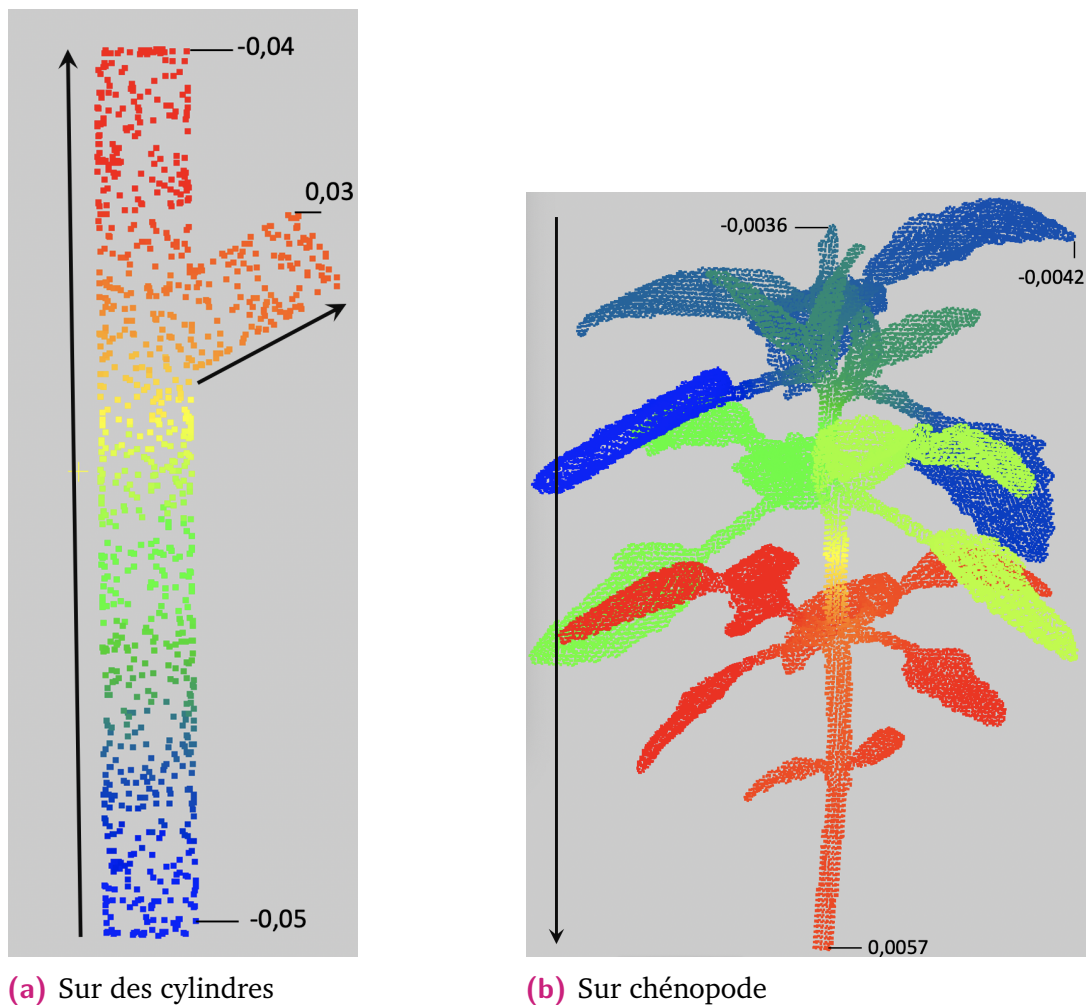
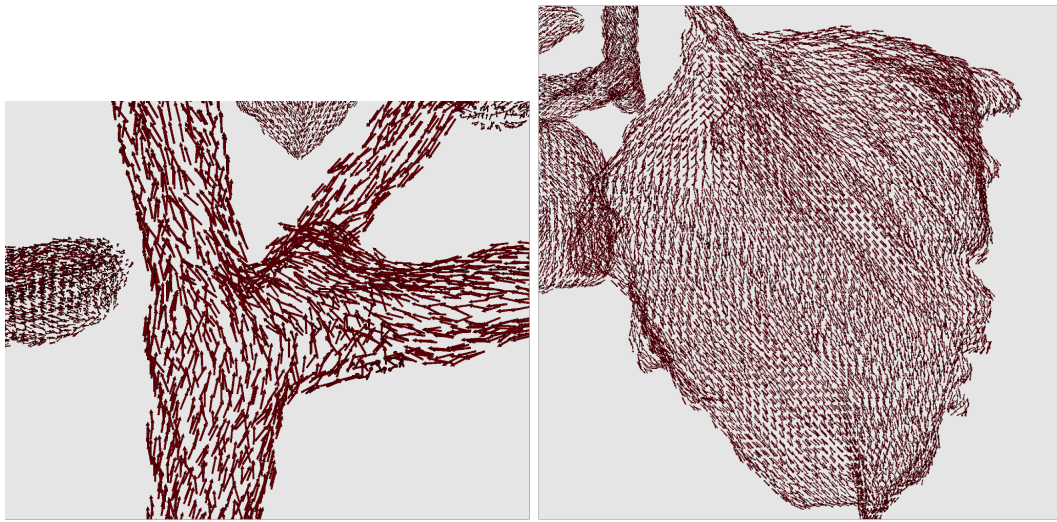


Figure 4.13: Vecteurs de Fiedler calculés sur deux nuages de points différents. Les valeurs du vecteur de Fiedler en chaque point sont représentées grâce à une échelle de couleur "Bleu->Vert->Jaune->Rouge". Quelques valeurs en certains points d'extrémités sont affichées sur les figures et les sens d'évolution des valeurs sont indiquées par des flèches.

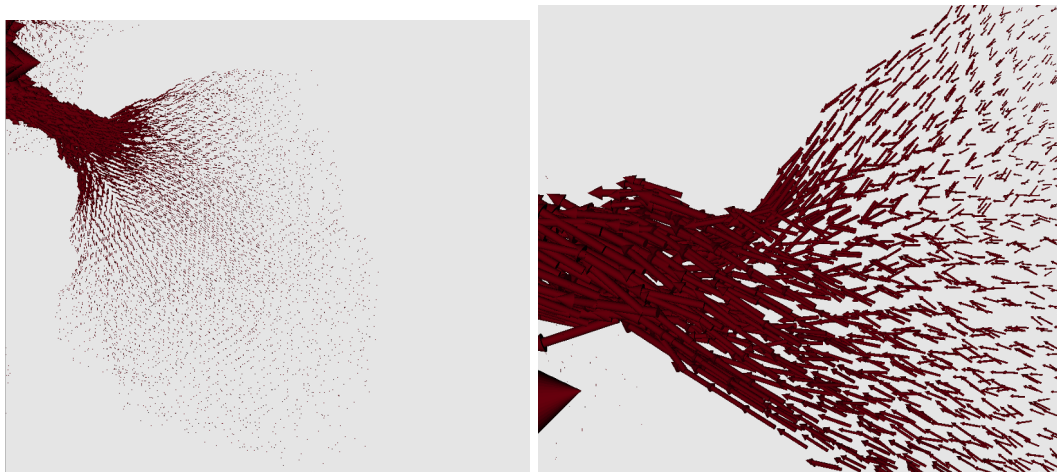
nœud du graphe. On remarque en figure 4.14a), que les directions du champ de vecteur suivent les organes linéaires de la plante. Des changements de directions importants ont lieu aux embranchements entre deux organes linéaires différents. A l'inverse, sur les feuilles, les directions divergent fortement (figure 4.14b). En figure 4.15, on peut constater que les valeurs de la norme diminuent de façon drastique entre les points d'un pétiole ou d'une branche et les points présents dans un apex ou un limbe.

On note en figure 4.16 que les normes au milieu de la tige principale sont plus importantes. Cette observation est cohérente avec la forme de la courbe du vecteur de Fiedler en chaque point (figure 4.11b)). La pente de la courbe est beaucoup plus forte en milieu de parcours qu'aux extrémités. On observe également un effet de bord,



(a) La direction des vecteurs est parallèle à la direction des organes linéaires. (b) Sur les feuilles, les directions divergent.

Figure 4.14: Champ de vecteurs normé calculé à partir du gradient du vecteur de Fiedler.



(a) La norme des vecteurs diminue fortement entre le pétiole et le limbe. (b) Zoom sur la diminution de la norme en entrée d'un limbe.

Figure 4.15: Champ de vecteurs calculé à partir du gradient du vecteur de Fiedler. La taille des traits est proportionnelle à la norme de chaque vecteur.

les normes diminuent fortement au niveau de la racine (bas de la tige principale). La racine de la tige peut donc, en terme de valeurs, être confondue avec les limbes et apex. Nous calculons donc le point le plus bas de la tige dans la suite ou le renseignons manuellement si des feuilles ont des coordonnées plus basses pour ne pas confondre l'instance dont il fait partie avec les limbes. A l'inverse, en haut de la tige, on observe le plus souvent un apex, l'effet de bord ne pose donc pas de problème pour la suite, puisque les petites normes correspondent aux autres apex/limbes présentes dans la plante.

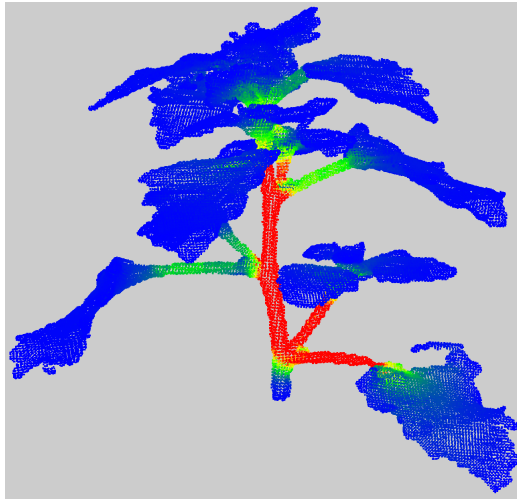


Figure 4.16: Nuage de points de chénopode réel issu de la chaîne de traitement ROMI. Chaque point est coloré selon la valeur de la norme du gradient du vecteur de Fiedler associé. L'échelle de couleur utilisée est "Bleu->Vert->Jaune->Rouge".

4.3 Algorithmes de segmentation

Dans la partie précédente, nous avons généré deux attributs géométriques permettant de différencier certains organes de la plante. D'une part les directions du GDM de Fiedler permettent de différencier les différents organes linéaires entre eux (tiges, pétioles, branches). D'autre part, la valeur de la norme du GDM pour les points appartenant à des limbes ou des apex est bien plus petite que pour le reste de la plante. L'objectif est dans cette partie d'exploiter ces attributs et d'obtenir une première segmentation par instance et une segmentation sémantique préliminaire.

4.3.1 Application de l'algorithme des K-means sur la norme

Alors que l'ensemble des directions permet de différencier un organe linéaire d'un autre, celles-ci ne semblent a priori pas permettre de discriminer les limbes des apex. On constate même que les directions ne sont pas homogènes et se dispersent, divergent sur les feuilles. Par conséquent, nous cherchons tout d'abord à exploiter la norme du GDM de Fiedler, de façon à isoler les limbes et les apex et à ne leur appliquer aucun traitement lié aux directions par la suite. C'est ce que l'on propose dans cette partie.

Nous avons observé que la norme du GDM de Fiedler est très faible sur les limbes et apex. Notre objectif est donc ici de réaliser une classe regroupant tous les points de très petite norme relativement à l'ensemble du nuage de points et qui correspondra au limbes et aux apex. Nous utilisons pour cela l'algorithme des K-means.

Algorithme classique de classification non supervisée : les K-means

L'algorithme des K-means est un algorithme devenu classique pour la segmentation non supervisée de données (MACQUEEN, 1967). Son principe consiste à séparer un ensemble de points en K groupes en fonction d'un critère donné. La variance au critère donné sera la même dans chaque groupe de points. Le nombre K d'instances voulues est donné par l'utilisateur. Dans l'algorithme classique des K-means (POLANSKI et KIMMEL, 2007) :

- On suppose donc que le nombre K de clusters est connu : il est donné par l'utilisateur.
- Chaque instance C est associée à un centroïde x_i^C , $i = 1, \dots, K$ dont les coordonnées sont égales à la moyenne des coordonnées des points appartenant à cette instance.
- Enfin, chaque point x de notre nuage, va être associé à un centroïde.

En entrée des K-means, il est donc nécessaire de donner un nombre de centroïdes, soit d'instances désirées. Or, lorsque l'on souhaite disposer d'un outil entièrement automatique, ce n'est pas possible. Dans notre cas, il ne paraît pas viable de demander à l'utilisateur de désigner le nombre d'instances qu'il cherche obtenir pour la plante étudiée. Il existe des méthodes permettant de déterminer automatiquement un nombre optimal de centroïdes d'initialisation.

Déterminer le nombre d'instances : la méthode du coude

La détermination du nombre d'instances à obtenir lors d'une segmentation de données est une problématique fréquente. Certains algorithmes tels que DBSCAN ou OPTICS évitent ce problème (ANKERST et al., 1999, ESTER et al., 1996). Cependant, ces algorithmes ne sont pas utilisables dans notre cas : les ruptures de valeurs de normes ne sont pas nettes mais progressives entre les limbes/apex et la partie linéaire. Les algorithmes basés sur la densité comme DBSCAN et OPTICS ne parviennent pas à repérer la rupture et ne divisent donc pas le nuage. Dans le cadre de l'algorithme des K-means, plusieurs options existent pour déterminer le nombre de instances. Les méthodes de détermination du nombre K consiste le plus souvent à calculer une segmentation pour différents K avec les K-means et à mettre en place un critère

d'évaluation de la qualité de la segmentation. La plupart des critères permettent de donner un score à la distance entre les points d'une classe (cohérence intra-classe) ou donnent un score à la distance entre les instances (différences inter-classes). Certains scores donnent un ratio en combinant ces deux notions. Ces indices sont des critères "internes" de qualité de la segmentation, ils ne nécessitent pas de segmentation de référence sur les données (critères évoqués dans le chapitre 3). On citera notamment l'indice de silhouette (ROUSSEUW, 1987) qui permet d'évaluer à la fois à le degré de proximité des points d'une même instance mais également à quel point chaque instance se distingue des autres. Cet indice est compris entre -1 et 1. La valeur de K qui sera retenue correspondra simplement à la valeur qui permet d'obtenir l'indice de silhouette le plus proche de 1. Néanmoins, avec cette façon de choisir K comme étant la valeur qui permet d'obtenir l'indice le plus élevé, on risque de toujours sélectionner le plus grand K testé. En effet, aussi pertinent que soit l'indice choisi, isoler des points avec des valeurs représentant des anomalies améliorera faiblement l'indice de segmentation. A l'inverse, il est possible si les transitions entre les points sont relativement douces, que le meilleur résultat soit toujours la valeur de K la plus petite.

On préfère donc généralement utiliser la méthode "du coude" (THORNDIKE, 1953) et qui peut s'appliquer à la plupart des critères d'évaluation internes. Cette méthode heuristique consiste à tracer la courbe représentant les valeurs du critère obtenues pour chaque segmentation réalisée avec les K -means en fonction du nombre K . On détermine ensuite le point de changement de pente ou "coude" de la courbe, la valeur de K après laquelle, même si l'indice s'améliore, l'ajout d'un nouveau centroïde ne représente pas une amélioration suffisante. Avant d'atteindre ce point de "coude", la valeur entre les indices calculés pour différents K peuvent augmenter rapidement, cela signifie que le modèle de segmentation est sous-ajusté. Après que le point de changement de pente ait été atteint, le modèle devient sur-ajusté, la pente d'amélioration entre les indices calculés est donc faible.

Pour l'exploitation de la norme nous avons utilisé l'indice de variance intra-classe, soit la moyenne des valeurs de variance en chaque classe. Autrement dit, nous ne nous intéressons qu'à la bonne proximité des points à l'intérieur de chaque instance et non pas à la distinction des instances entre elles.

La méthode appliquée sur nos plantes avec l'attribut de la norme

Nous appliquons les K -means afin de séparer une grande partie des apex/limbes des organes linéaires de la plante. On notera que pour cette première étape, nous n'appliquons plus la méthode du coude, l'expérience, et la méthode du coude nous a

montré que sur les plantes dont nous disposons, demander en entrée, 4 clusters est optimal pour segmenter au mieux les limbes/apex. Cela nous permet d'économiser en temps de calcul. Néanmoins, il est toujours possible d'appliquer la méthode du coude sur un nouveau type de données pour déterminer un nombre K automatiquement.

On obtient une première segmentation globale de la plante, comme présentée en figure 4.17. Sur cette figure, on peut constater le résultat de cette segmentation préliminaire sur trois espèces de plantes différentes dont les modèles sont issus de processus d'acquisition différents. Le chénopode présenté est issu de la chaîne de traitement ROMI (section 2.3.1), le peuplier est une jeune pousse, scannée par Lidar (selon la méthode de CASELLA et SINOQUET, 2003, il s'agit d'un peuplier du jeu de données générées dans HÉTROUY-WHEELER et al., 2016), le plant de tomates est issu d'un traitement photogrammétrique minimaliste (photos prises avec un smartphone puis reconstruction via photoscan (AGISOFT, 2014)). Les nuages de points ont donc des types de bruits différents (BERGER et al., 2017). Alors que le chénopode présente un échantillonnage uniforme et relativement peu bruité, le plant de tomate dispose d'un échantillonnage non uniforme, bruité, avec des points aberrants liés à l'arrière plan mal filtré. Basé sur le spectre du graphe, on constate ici que la norme du GDM de Fiedler est robuste au bruit, comme attendu selon la théorie.

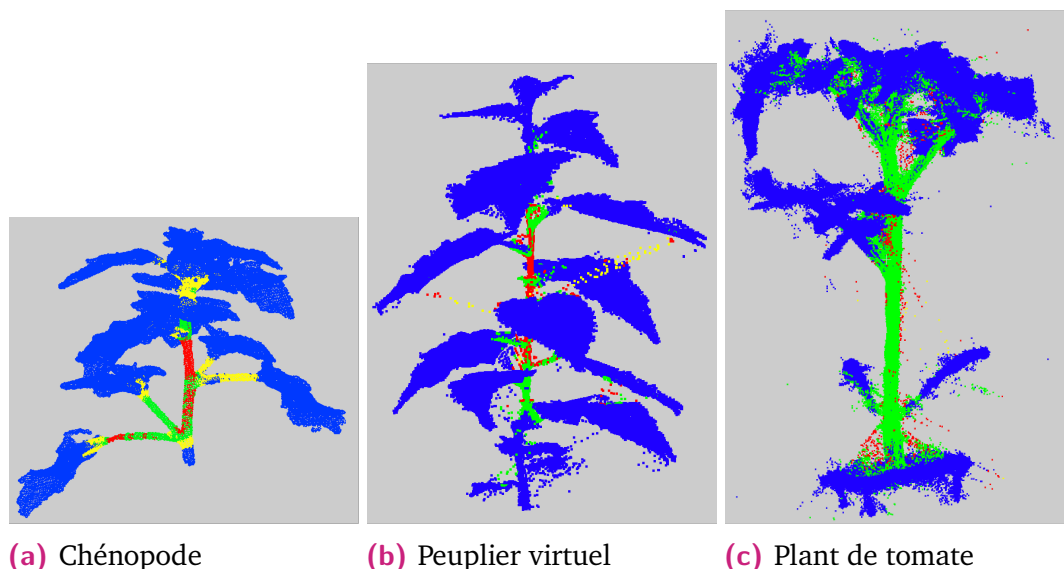


Figure 4.17: Résultats suite au calcul du gradient du vecteur de Fiedler et à l'exploitation de sa norme par l'application de l'algorithme des K -means avec $K=4$. La partie bleue correspond à la classe rassemblant les points portant les normes les plus petites. Il correspond donc aux limbes et apex.

4.3.2 Algorithmes de croissance de région pour l'obtention d'une segmentation par instance

Lorsque l'on parvient à rassembler tous les organes de même type d'une plante dans la même classe, nous nous rapprochons d'une segmentation sémantique de la plante. Avec les algorithmes des K-means, tous les apex et les limbes sont regroupés dans la même classe comme on peut le voir en figure 4.17. Néanmoins, nous cherchons également à obtenir une segmentation par instance de la plante, c'est à dire à identifier chaque organe individuellement. L'objectif est donc de séparer les éléments qui ne sont pas connexes dans le graphe et de les associer à une instance chacun. Afin de séparer chaque limbe et chaque apex dans une instance différente, nous pouvons faire appel à un algorithme de croissance de région. L'algorithme de croissance par régions est l'algorithme que nous avons utilisé dans notre chaîne de traitement pour obtenir une segmentation par instance à partir d'une segmentation sémantique. Cet algorithme consiste à sélectionner un nœud du graphe (qui sera appelé graine) de façon aléatoire puis d'ajouter dans la même région ou instance tous ses voisins sur le graphe qui ont la même propriété (ici la même classe obtenue grâce aux k-means). Tous les nœuds nouvellement ajoutés à la région de la première graine, deviennent des graines à leur tour dont on parcourt le voisinage. Une fois qu'il n'y a plus de nœuds à agglomérer dans cette région, on tire un nouveau nœud du graphe qui n'a pas encore été regroupé dans une instance. Une version en pseudo-code de cet algorithme et de la façon dont il a été implémenté est décrite ci-dessous.

Le résultat est présenté en figure 4.18. On notera que cet algorithme ne permet pas de séparer des limbes ou des apex qui seraient connectés dans le graphe de similarités. En effet, si deux limbes se touchent, ils se retrouveront dans la même instance ou région de l'algorithme. D. LI, CAO et al. (2019) propose une solution pour la séparation des limbes qui se touchent.

Algorithm 1: Region growing on the similarity graph to obtain instance segmentation P

Require: Segmented similarity graph $G = (V, E, C)$ with $\text{card}(V) = n$ and every vertex v is associated to a class $C(v)$

Ensure: Segmentation of the ensemble of nodes V into a connected component P_i such as $\cup_{i=1}^p P(i) = V$ with p the number of parts obtained. Every vertex v is associated to a part $P(v)$

$visited :=$ list of n entries initialized at 0

$queue :=$ empty list

$p := 0$

for $i \in V$ and $visited[i] == 0$ **do**

$visited[i] = 1$

 Add i to list $queue$

while $queue$ is not empty **do**

$j = queue[0]$

 Remove j of the list $queue$

$P(j) = p$

for Every neighbor v of the vertex j in G **do**

if $C(v) == C(j)$ and $visited[v] == 0$ **then**

 Add v to the end of list $queue$

$visited[v] = 1$

end if

end for

end while

$p += 1$

end for

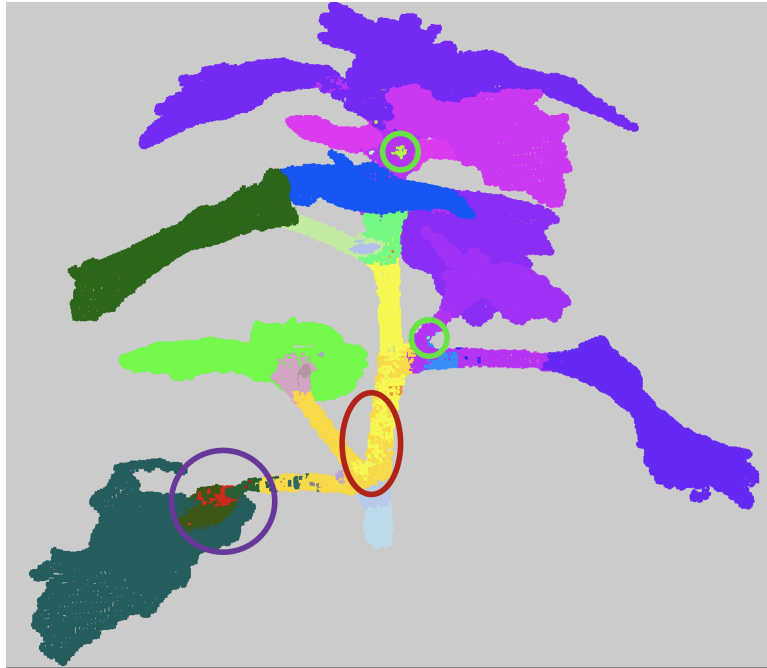


Figure 4.18: Résultat de la segmentation par instance. Le nuage de points a été pré-segmenté grâce au K-means sur la norme du GDM de Fiedler puis a subi un traitement de croissance de région. Il est donc segmenté par instance, chaque couleur représente une instance. On notera que chaque limbe est maintenant dans sa propre instance. L'algorithme a donc accompli sa mission. Néanmoins, on note également la présence de défauts fréquents suite à l'utilisation d'un algorithme ne prenant pas directement en compte la topologie du graphe de similarité construit ou les coordonnées des points du nuage. Les cercles verts montrent des instances isolées et mal segmentées. On peut voir dans le cercle rouge deux instances mélangées entre elles. Enfin, le cercle violet montre des frontières mal définies entre les instances présentes au bas de la feuille de ce modèle de chénopode.

4.3.3 Outil de filtrage de la segmentation

Sursegmentation et besoin de filtrage

Lorsque l'on applique un algorithme tel que les K-means ou tout autre algorithme basé sur la minimisation de la variance intra-classe d'un attribut de chaque point, on ne prend pas forcément en compte la topologie du graphe de similarité sous-jacent ou les coordonnées dans l'espace des points représentés par les nœuds du graphe. Il se peut également que l'attribut soit bruité et entraîne une mauvaise classification de certains points qui sont isolés topologiquement des autres points disposant des mêmes valeurs d'attribut. On constate que certaines classes sont mélangées spatialement sur le graphe, ou que les frontières entre les classes sont parfois floues.

Lorsque l'on applique l'algorithme de croissance par régions, on reprend en compte la topologie du graphe et on fragmente les classes en sous-instances. Cela nous permet de récupérer des instances pertinentes : les limbes sont à présent bien séparés. Néanmoins, des points ou groupes de points sont isolés et ne sont pas pertinents. La figure 4.18 illustre les différents types d'erreur que l'on peut rencontrer suite à une classification sans prise en compte de la position des points dans le nuage mais uniquement selon l'attribut d'intérêt choisi. Suite à une telle segmentation, il peut être intéressant de disposer d'un outil permettant la résolution de ces erreurs. En effet, outre la mauvaise définition des frontières des instances, la présence de nombreuses petites instances peut augmenter le temps de calcul de certains traitements que l'on verra par la suite.

Définition d'une énergie

Pour résoudre ce problème de sursegmentation, nous proposons une approche basée sur une optimisation d'énergie inspirée des modèles de Potts (Loic LANDRIEU et SIMONOVSKY, 2018). Nous souhaitons définir une énergie pour chaque nœud de notre graphe de similarité. Nous travaillons sur un nuage de points segmenté en plusieurs instances et sur une structure de graphe de similarité construite en section 4.1.1. Nous travaillons donc sur le graphe G segmenté en instances :

$$G = (V, E, P)$$

V : ensemble des nœuds de G

E : ensemble des arêtes de G

P : segmentation par instance, en parties de G

On distingue deux types d'arêtes, telles que $E = E_0 \cup E_1$. E_0 est l'ensemble des arêtes intraclasse et E_1 est l'ensemble des arêtes interclasse. Autrement dit, E_0 regroupe toutes les arêtes du graphe qui lient deux nœuds ayant été segmentés dans la même instance. En notant $P(u)$ l'instance du nœud u :

$$\forall (u, v) \in E, \quad (u, v) \in E_0 \quad \text{ssi} \quad P(u) = P(v)$$

$$\forall (u, v) \in E, \quad (u, v) \in E_1 \quad \text{ssi} \quad P(u) \neq P(v)$$

Améliorer la segmentation revient ici à obtenir un ensemble d'arêtes interclasse le plus petit possible. On définit donc notre énergie sur une arête comme suit :

$$\begin{aligned} \mathcal{E}((u, v)) &= 0 \quad \text{si} \quad (u, v) \in E_0 \\ &= 1 \quad \text{si} \quad (u, v) \in E_1 \end{aligned} \tag{4.10}$$

Il nous est alors possible de calculer une énergie en chaque nœud. Intuitivement, nous aimerions l'écrire simplement comme une somme de l'énergie de chaque arête incluant le nœud d'intérêt. $\forall v \in V$ et en notant $N(v)$ l'ensemble des nœuds dans le voisinage de v :

$$\mathcal{E}_{int}(v) = \frac{\sum_{u \in N(v)} \mathcal{E}(v, u)}{|N(v)|}$$

L'énergie en un nœud est normalisée entre 0 et 1. Lorsqu'un nœud a une énergie inférieure à 0.5, cela signifie qu'il est majoritairement entouré de nœuds se trouvant dans la même instance que lui-même, comme en figure 4.19a). A l'inverse, lorsque son énergie est supérieure à 0.5 comme en figure 4.19b), cela signifie que le nœud est entouré d'instances différentes. Enfin, lorsque l'énergie topologique du nœud est égale à 0,5, c'est qu'il y a autant de nœuds dans la même instance que de nœuds dans une instance autre, comme en figure 4.19c). Enfin, si le nœud à l'étude est entouré de nœuds appartenant uniquement à la même instance que lui, son énergie est nulle (figure 4.19d)). Néanmoins, il existe un cas problématique qui n'est pas géré par cette formule immédiate. En effet, lorsque plus de deux instances se trouvent dans le voisinage du nœud comme en figure 4.19e), bien que l'état d'équilibre soit atteint, on n'obtient pas une énergie de 0.5 mais une énergie supérieure montrant que le cas de figure est défavorable. Autrement dit, avec cette formule on obtient une énergie qui suggère que changer le nœud d'instance permettrait de diminuer son énergie. Ce n'est pas le cas ici, un équilibre a été atteint.

Les états d'équilibres possibles à partir de trois instances voisines ne sont donc pas pris en compte. La formule a donc été adaptée pour ne pas prendre en compte uniquement le nombre de nœuds voisins mais également le nombre d'instances différentes présentes dans le voisinage.

$$\mathcal{E}(v) = \frac{\sum_{u \in N(v)} \mathcal{E}(v, u)}{e_1(v) + (p(v) - 1) * e_0(v)} \quad (4.11)$$

Avec :

$e_1(v)$: nombre d'arêtes connectées à v telles que $\forall u \in N(v) \quad (u, v) \in E_1$, également égale à $\sum_{u \in N(v)} \mathcal{E}(v, u)$ telle que définie en equation 4.10

$p(v)$: nombre d'instances présentes dans le voisinage de v

$e_0(v)$: nombre d'arêtes connectées à v telles que $\forall u \in N(v) \quad (u, v) \in E_0$

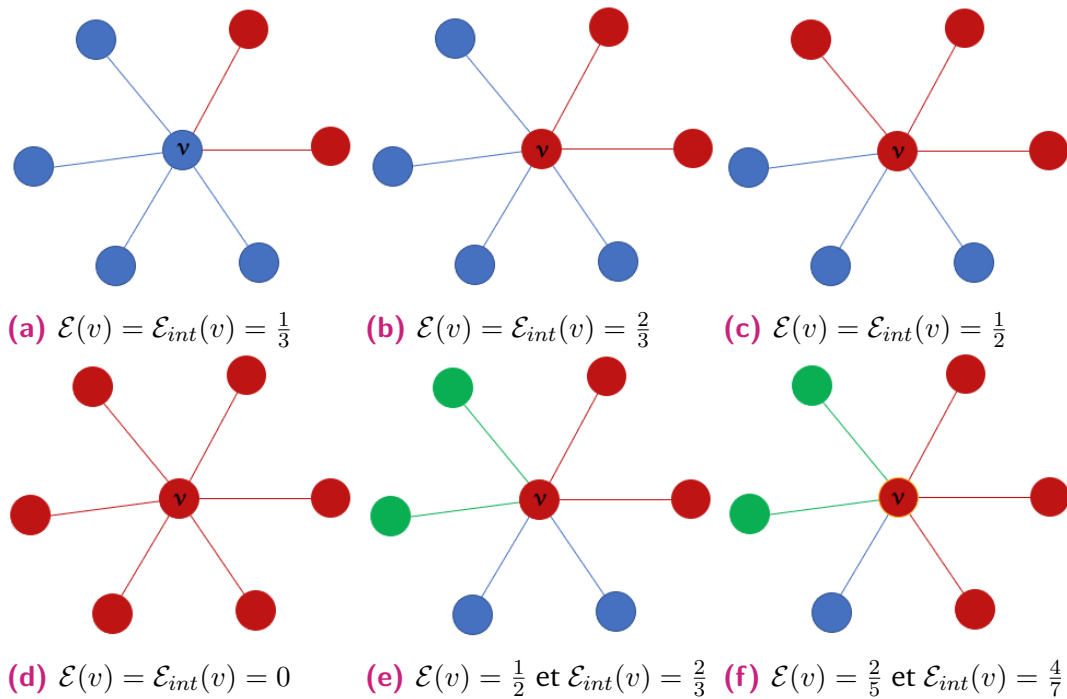


Figure 4.19: Différents cas de figure pour le calcul de l'énergie topologique en un nœud v , valeur de l'énergie avec la formule intuitive $\mathcal{E}_{int}(v)$ et la formule finale utilisée $\mathcal{E}(v)$.

Comme on peut le voir en figure 4.19e) et 4.19f), la formule finale permet de gérer les cas comprenant plusieurs instances voisines mais ne change pas le résultat lorsque seules deux instances sont en contact.

Optimisation de l'énergie

Nous calculons les énergies de tous les nœuds appartenant au graphe G . Nous sélectionnons un nœud v de façon aléatoire parmi les nœuds de haute énergie $\mathcal{E}(v) > 0.5$. Nous effectuons un tirage aléatoire pondéré parmi les instances présentes dans le voisinage de v . L'instance la plus prépondérante dans le voisinage a ainsi plus de chances d'être sélectionnée. Le nœud est associé à cette autre instance et son énergie, ainsi que celle de ses voisins sur le graphe de similarité sont recalculées. Cette opération est réitérée jusqu'à ce qu'un équilibre soit atteint au niveau de tous les nœuds en frontière.

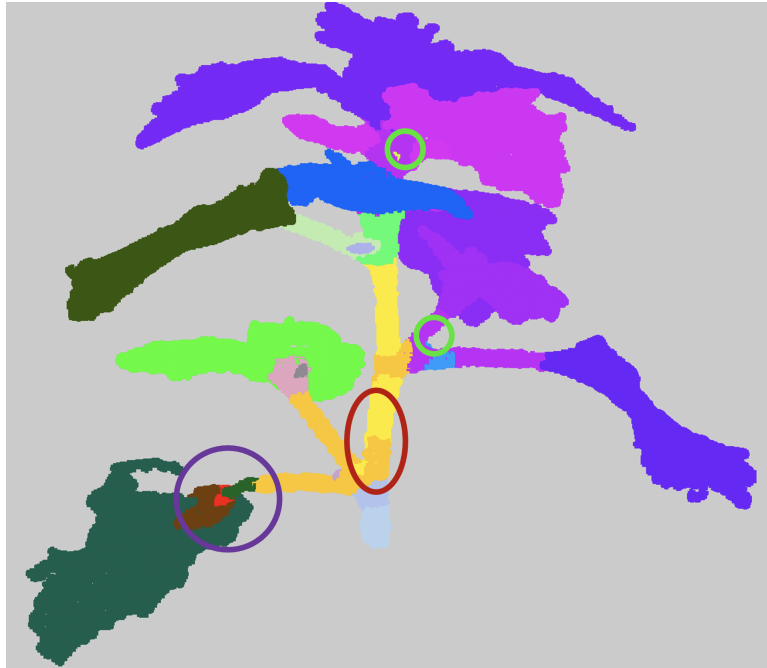


Figure 4.20: Défauts de segmentation résolus suite à l'application du filtre basés sur l'énergie topologique en chaque point.

Résultats et discussion sur le filtrage topologique

A l'issue de ce traitement, les défauts énumérés en section 4.3.3 ont été considérablement diminués, comme on peut le constater en figure 4.20. Les frontières sont plus nettes, les petites instances ont été réduites ou supprimées.

Ce filtrage ne garantit pas le respect de la segmentation existante. Il s'agit d'un traitement uniquement topologique sur le graphe quotienté. Le résultat visuel suite à ce filtrage est dépendant de la régularité du nuage de points. Si les points sont disposés en grille, le résultat qualitatif semblera de meilleure qualité.

4.4 Conclusion

Afin de calculer des attributs pertinents pour regrouper les points en fonction des organes qu'ils représentent. Afin d'éviter une difficulté inhérente au partitionnement spectral qui consiste à déterminer le bon nombre de vecteurs propres à utiliser pour l'objet traité nous avons, après observations, utilisé le vecteur de Fiedler. Nous avons calculé un champ de vecteur issu de ce dernier. Les normes du GDM du vecteur de Fiedler diminuent drastiquement entre les pétioles/branches et les apex/limbes. Cette caractéristique nous permet avec l'emploi des K-means de séparer les apex/limbes du

reste de la plante. Néanmoins, l'utilisation des K-means ne prend pas en compte la topologie du graphe adjacent. Par conséquent tous les apex/limbes sont dans la même instance après cet algorithme et on note également que les instances ne sont pas très bien séparées entre elles sur le graphe. Pour résoudre ce problème nous avons proposé deux algorithmes, d'une part un algorithme de croissance par région pour récupérer une instance différente pour chaque apex ou limbe de la plante. D'autre part, nous proposons un algorithme de filtrage original basé sur un concept de minimisation d'énergie topologie en chaque point.

A l'issue de ces étapes nous obtenons donc une première segmentation par instance disposant de deux classes : les limbes/apex et les instances linéaires. Néanmoins des erreurs persistent, il arrive que des ensembles pétioles et limbes soient sous segmentés et donc réunis dans la même instance. De plus, les directions du GDM du vecteur de Fiedler sont difficilement exploitables à l'échelle du graphe de similarité, puisque l'on cherche à rassembler les éléments disposant en moyenne des mêmes directions. Celles-ci étant bruitées et floues dans les zones frontières, une approche point à point n'est pas envisageable.

Les erreurs de segmentation semblent inévitables lorsque l'on s'appuie sur des attributs purement géométriques ou même lorsque l'on prend une approche basée sur l'apprentissage des réseaux neuronaux (voire chapitre 3). De plus, la segmentation de nuages de points en prenant en compte le contexte des points de façon plus large que leur simple voisinage est également une problématique fréquente dans le domaine (voire chapitre 3) Deux besoins apparaissent donc, la possibilité de traiter des ensembles de points et l'utilisation de connaissances botaniques pour détecter les erreurs de segmentation.

Introduction de connaissances botaniques et changement d'échelle

Lorsque l'on traite les imperfections et défauts d'une segmentation, il peut arriver que traiter chaque point du nuage ne soit pas la bonne approche. Dans certains cas, on souhaite travailler avec des groupes de points et l'emplacement topologique de ces derniers les uns par rapport aux autres. L'approche purement géométrique basée sur la forme du nuage de points a également certaines limites que nous avons pu aborder dans le chapitre précédent. Des connaissances botaniques sur les plantes permettraient de compléter et de potentiellement corriger les erreurs présentes dans la segmentation par instance. De plus, l'utilisation de connaissances botaniques semble indispensable pour classer les instances dans des classes d'organes.

L'objectif est ici d'introduire un nouvel objet, permettant de travailler non plus à l'échelle du point mais à l'échelle d'une partie. Dans notre cas, nous souhaitons *in fine* que chaque partie du nuage soit un organe identifié de la plante. La nouvelle échelle va donc nous permettre d'une part de travailler avec une nouvelle topologie mais également d'introduire des connaissances botaniques, plus précisément des connaissances sur la hiérarchie des axes et organes de la plante et les relations entre ses organes. Ces deux aspects sont complémentaires dans notre approche. La nouvelle topologie va nous permettre de trouver la sémantique et celle-ci nous permettra de corriger la segmentation sémantique sous-jacente.

Dans une première partie nous introduisons cet objet, le graphe quotient, et expliquons comment nous le construisons à partir d'une segmentation par instance préexistante. Nous nous intéressons ensuite aux algorithmes permettant d'exploiter les directions du GDM du vecteur de Fiedler, puis aux algorithmes permettant la classification des différentes instances et enfin nous proposons une façon d'envisager la correction de la segmentation par instance.

5.1 Le graphe quotient : changement d'échelle dans l'exploitation d'attributs

5.1.1 La construction d'un graphe quotient

On cherche à construire à partir du graphe de similarité G segmenté (que l'on appelle graphe quotienté), un graphe quotient \mathcal{Q}_g .

Le graphe de similarité $G = (V, E)$ défini en section 4.1.1 avec son ensemble de nœuds V et son ensemble d'arêtes E a été augmenté lors des opérations suivantes (sections 4.3.1 et 4.3.2) d'une segmentation par instance telle que $G_s = (V, E, P)$ avec $P = \{V_1, V_2, \dots, V_p\}$ l'ensemble des p instances créées. A partir du graphe de similarité segmenté G_s , on construit un graphe $\mathcal{Q}_g = (\mathcal{V}, \mathcal{E})$, avec \mathcal{V} son ensemble de nœuds et \mathcal{E} son ensemble d'arêtes. Chaque nœud du graphe quotient représente une instance du graphe de similarité segmenté, soit $\mathcal{V} = P$ donc $\text{card}(\mathcal{V}) = p$. Il existe une arête $(V_1, V_2) \in \mathcal{E}$ si et seulement si il existe une arête $(v_1, v_2) \in E$ telle que $v_1 \in V_1$ et $v_2 \in V_2$.

Cette opération est décrite en figure 5.1. La première étape représente le graphe de similarité sur une plante réelle (*chenopodium album*). Il s'agit de la topologie microscopique disponible jusqu'à présent, autrement dit, il s'agit du graphe de similarités G . D'autre part, chaque point du nuage de points s'est déjà vu attribuer une étiquette ou une couleur lors de traitements précédents, désignant dans quelle partie il se trouve. La deuxième étape est représentée par un schéma dans la figure 5.1. Une portion du graphe de similarité segmenté est représentée ici, ce même objet dispose des informations topologiques et des étiquettes de segmentation pour chaque nœud. Le passage du graphe quotienté au graphe quotient consiste à créer un nouveau graphe. Chaque nœud de ce nouveau graphe représente une partie connexe sur le graphe de similarité. L'ensemble des nœuds dans la partie "orange" seront représentés par un seul nœud dans le graphe quotient. Si deux parties, par exemple la partie orange et la partie bleue, sont voisines dans le graphe de similarités, nous retrouverons une arête entre les deux nœuds représentant ces parties dans le graphe quotient.

Le graphe ainsi construit à partir de notre première segmentation géométrique présente en général des cycles, 5.1b). Ces cycles peuvent représenter une réalité géométrique. En effet, même lorsque l'on dispose d'une segmentation de référence, il est très possible qu'une branche et un pétiole soient très proches au niveau de leur

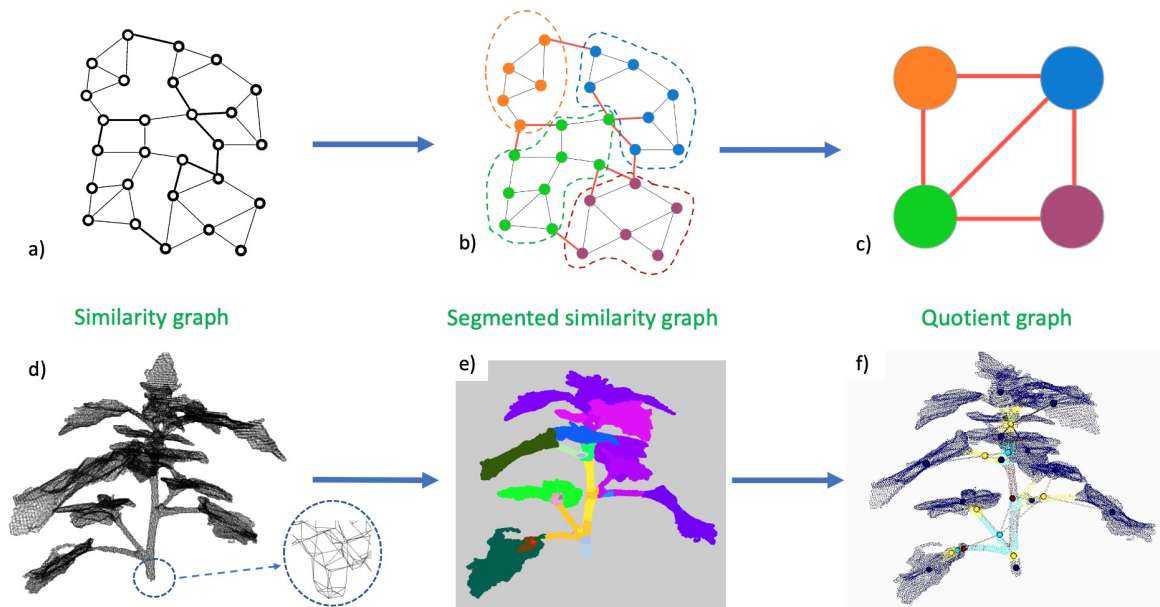


Figure 5.1: a)-b)-c) Schéma de construction d'un graphe quotient. La première étape Le nuage de points segmenté par instance (de façon encore imparfaite) associé est avec le même angle de vue est affiché en dessous. La deuxième étape est représentée par un schéma reprenant une partie de la structure de graphe de similarité qui aurait été segmenté en 4 parties. Le graphe quotient est représenté dans la troisième partie. Chaque partie connexe du graphe de similarité est représentée par un nœud dans le graphe quotient. d) Représente le graphe de similarité sur une plante réelle (*chenopodium album*). Un zoom a été effectué pour permettre de constater de la structure de graphe de l'objet. e) Segmentation par instance sur un plant de chénopode f) Superposition d'un nuage de points de chénopode et de son graphe quotient associé, les points bleus appartiennent à la classe limbe/apex, les autres couleurs sont pour les parties linéaires non classés. Les points du graphe quotient ont été placés en 3D en calculant les coordonnées moyennes de l'ensemble des points de la partie qu'ils représentent. Ils sont de la même couleur que la classe de la partie qu'ils représentent.

embranchement avec la tige principale. Les points du graphe de similarité forment alors un cycle à cet endroit, alors que d'un point de vue botanique, on souhaiterait plutôt que le pétiole et la branche soient représentés comme étant uniquement reliés à la tige principale. Notre objectif pour le graphe quotient dans la suite de ce chapitre consiste donc à le transformer en un arbre (sémantique) représentatif des relations entre les organes de la plante. Concrètement, il s'agit de transformer le graphe quotient de façon à ce qu'il devienne un arbre au sens informatique du terme dont chaque nœud représente un organe de la plante et chaque arête correspond à une connexion entre organes botaniques. Pour définir une arborescence (arbre orienté), on peut d'ors et déjà identifier le point le plus bas de la tige puisque l'utilisateur l'a renseigné ou le point a été calculé (section 4.2.3 et désigner le nœud de l'arbre quotient auquel il appartient comme étant le nœud "racine" du graphe quotient ce qui permet d'introduire une hiérarchie et de se rapprocher d'un arbre orienté.

5.1.2 Notes sur les attributs exploités

Une fois le graphe quotient construit, il est possible d'associer à ses nœuds ou à ses arêtes des attributs. Certains attributs pourront par exemple relever de sa construction. Il est par exemple possible de compter le nombre de connexions au niveau du graphe de similarité, entre plusieurs points, qui sont à l'origine d'une arête entre deux nœuds du graphe quotient et d'attribuer cette valeur à l'arête (nombre de connexions inter classes). De la même façon on peut attribuer à un nœud du graphe quotient le nombre de points présents dans la partie qu'il représente ou encore le nombre d'arêtes présentes entre les nœuds de la partie (nombre de connexions intra classe). De plus, tous les attributs calculés et attribués à des nœuds du graphe de similarité sont moyennables et il est possible de les attribuer au nœud qui représente la partie dont font partie ces points dans le graphe quotient.

Il est également possible de calculer et de stocker simplement des informations géométriques qui seraient calculées sur l'ensemble des points d'une partie. Nous en verrons un exemple en section 5.2.2.

5.2 Les algorithmes sur la structure du graphe quotient

Nous présentons ici l'ensemble des algorithmes développés qui utilisent la structure du graphe quotient afin d'améliorer la segmentation par instance et la classification sémantique. Toute amélioration de la segmentation par instance ou sémantique se répercute sur la structure du graphe quotient. Au fil de ces algorithmes, nous allons modifier le graphe quotient et obtenir un arbre contenant des informations sémantiques, que nous appellerons "arbre quotient sémantique".

Grâce aux algorithmes précédents, nous avons pu déterminer géométriquement quelles sont les parties linéaires de la plante et les séparer de l'ensemble foliaire (apex/limbe). Néanmoins, parmi les éléments que l'on peut qualifier de linéaires, on peut trouver différents organes : les pétioles, la tige principale, les branches. De la même façon, si l'on est parvenu à distinguer les apex et les limbes des feuilles des organes linéaires, les distinguer entre eux est plus délicat.

5.2.1 Division/fusion de nœuds : l'exploitation des directions

Afin d'obtenir une segmentation plus fine des éléments linéaires, nous exploitons les directions du gradient du vecteur de Fiedler. Pour cela nous utilisons une approche qui consiste à diviser puis fusionner ("split and merge"). Dans un premier temps, il s'agit de s'assurer que chaque partie est homogène en terme de directions. En d'autres termes, on souhaite s'assurer pour une plante que nous n'avons pas dans la même partie un morceau de tige avec un morceau de pétiole par exemple. Ce cas de figure est présenté en figure 5.3a par les parties violettes et turquoises sur la tige. Pour s'assurer de cette homogénéité, chaque partie est resegmentée en fonction des directions du gradient du vecteur de Fiedler. La méthode utilisée que nous avons choisie est celle des K-means avec la méthode du coude, présentée en section 4.3.1. Notons néanmoins que toute méthode de classification non supervisée qui n'aurait pas en entrée le nombre de parties désirées peut être envisagée pour cette tâche. A l'échelle du graphe quotient, cela signifie que l'on augmente le nombre de nœuds de ce dernier, comme représenté sur les deux premiers schémas dans la figure 5.2.

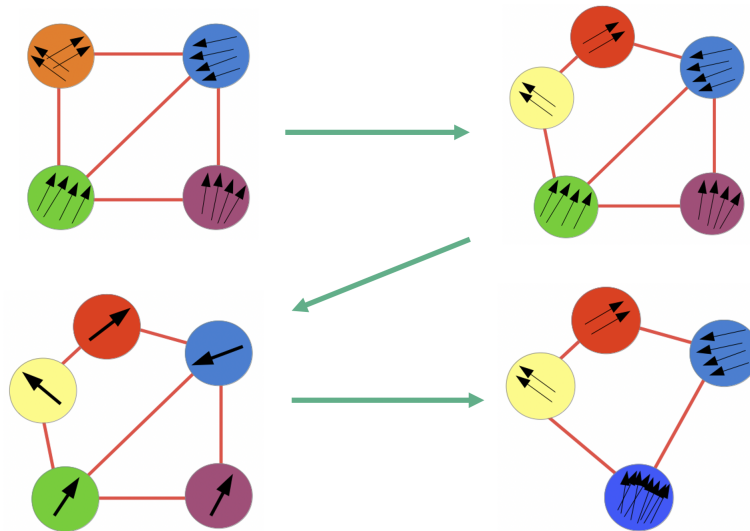
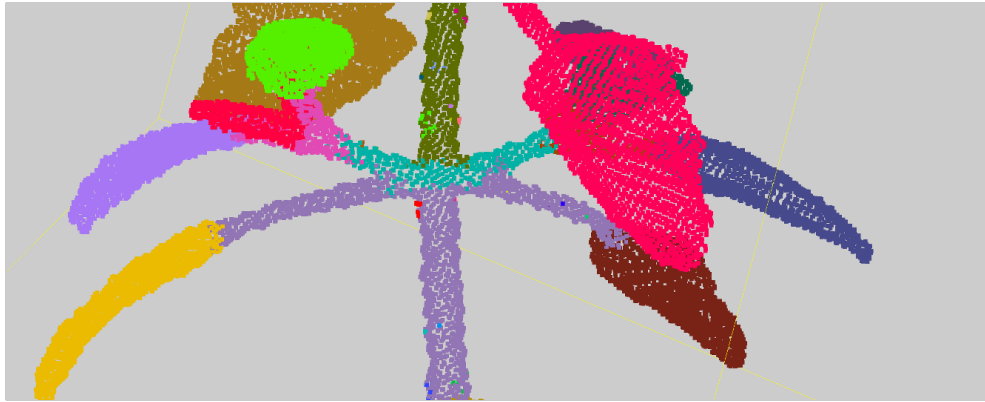


Figure 5.2: Schéma descriptif de la méthode de division/fusion

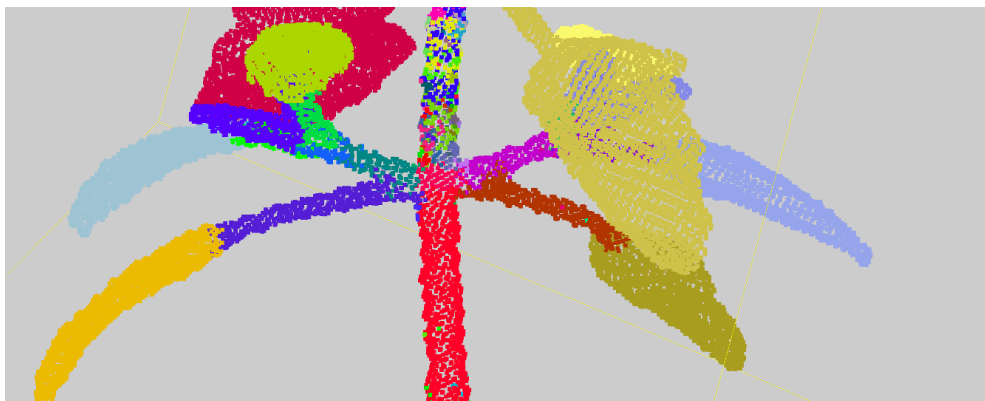
Lorsque la première étape de division a été réalisée, nous nous situons dans le cas de figure présenté en figure 5.3b. Il est très possible que certaines parties soient sur-segmentées du fait du bruit présent sur les directions, ici c'est le cas de la partie haute de la tige. Nous ajoutons donc une étape de fusion. Nous calculons la moyenne des directions normalisées v_i de chaque partie. Ainsi, chaque nœud i du graphe quotient dispose d'un nouvelle attribut : la direction moyenne \bar{D}_i des vecteurs au nœud i . L'objectif est maintenant de déterminer quels sont les nœuds adjacents dont les directions moyennes sont assez proches pour être fusionnées. Pour cela, pour chaque arête entre deux nœuds, on calcule une énergie $E_{i,j}$ basée sur le produit scalaire entre deux vecteurs, préalablement normalisés, telle que :

$$E_{i,j} = 1 - \bar{D}_i \cdot \bar{D}_j \quad (5.1)$$

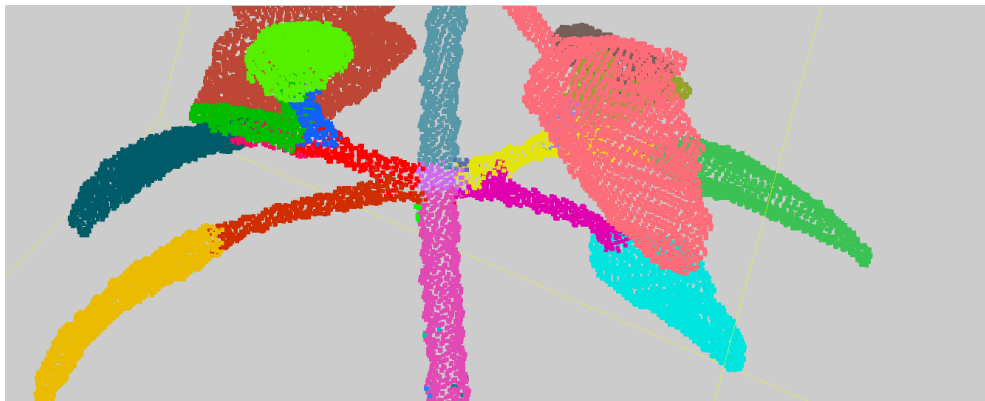
Cette formulation permet de définir une énergie positive proche de 0 si deux vecteurs sont colinéaires, autrement dit aux arêtes dont les nœuds peuvent être fusionnés. L'arête portant l'énergie la plus basse est donc supprimée et les deux nœuds du graphe quotient à ses extrémités sont fusionnés en même temps que les deux parties du graphe de similarité qu'ils représentent. Cette étape est réitérée sur les nœuds du graphe quotient jusqu'à ce qu'un seuil d'énergie soit atteint. Nous avons déterminé expérimentalement que sélectionner un angle de 30° , c'est-à-dire un seuil sur $E_{i,j}$ de $1 - \sqrt{3}/2$ entre \bar{D}_i et \bar{D}_j donnait des résultats satisfaisants pour la suite de notre algorithme. Néanmoins, ce seuil peut dépendre de la densité du nuage de points et de la taille du voisinage considérée pour le calcul du gradient du vecteur de Fiedler.



(a) Segmentation par instance comportant des défauts au niveau des parties linéaires.



(b) Segmentation par instance après application de l'algorithme de division.



(c) Segmentation par instance après application de l'algorithme de fusion.

Figure 5.3: Action de l'algorithme de division/fusion sur la segmentation du nuage de points. Le nuage de points utilisé ici est un nuage de points de chénopode réel. Les points sont colorés grâce à la segmentation par instance réalisée. Chaque couleur représente une instance.

Finalement, cette approche permet d'obtenir des résultats comme présentés en Figure 5.3c. Les parties linéaires sont segmentées en parties plus cohérentes géométriquement et pour le traitement des organes par la suite. On notera néanmoins que la partie de division de cet algorithme introduit des artefacts, il est possible de les

filtrer avec les algorithmes présentés en section 4.3.3, mais une partie de ces petites parties sera automatiquement filtrée dans la suite.

5.2.2 Maxima locaux : la différenciation apex/limbe

Le chapitre précédent nous a montré comment déterminer quels étaient les nœuds pouvant être des apex ou des limbes, ils ont été mis dans une même classe d'objets. On souhaite maintenant les distinguer les uns des autres, disposer d'une classe limbe et d'une classe apex. L'algorithme 2 décrit en pseudo-code les opérations mises en place pour réaliser cette étape. Pour effectuer cette opération, on utilise les valeurs du vecteur de Fiedler en chaque point du graphe de similarité. Étant donnée nos observations en Section 4.2, un local minimum ou maximum est atteint dans les valeurs du vecteur aux extrémités du nuage de points. On recherche donc les extrema locaux du vecteur de Fiedler dans le nuage. Pour calculer ces extrema locaux, nous avons utilisé le graphe de similarités. En chaque nœud v du graphe de similarité, on considère tous les nœuds de son voisinage et on compare la valeur du vecteur de Fiedler au nœud v , $f(v)$, avec tous ses voisins. Si $f(v)$ est la plus petite (ou la plus grande) valeur du vecteur de Fiedler, cela signifie que le nœud v représente un extremum local. Aucun nœud ne porte de valeur plus petite (ou grande) dans cette zone du graphe de similarité. Le nombre d'extrema locaux est alors transmis comme un attribut au graphe quotient. Chaque nœud classé en limbe/pétiole dispose d'un nombre de maxima locaux. Si le nombre d'extrema locaux est égal à 1 (figure 5.4a), le nœud est considéré comme un limbe, autrement s'il est supérieur, c'est un apex (figure 5.4b)).

Cette approche est efficace pour des feuilles de forme simple. Dans le cas de feuilles dentelées ou crénelées (ou irrégulière au niveau de leur bordure, voir chapitre 2) visibles en figure 5.4c), il peut être nécessaire de prendre un voisinage plus grand pour éviter de considérer ces irrégularités du limbe comme des extrema locaux. Cela vaut aussi si le nuage de points est particulièrement irrégulier sur les bords, ou que l'on constate des défauts d'acquisitions (BERGER et al., 2017). Enfin, dans le cas des feuilles composées ou simplement de feuilles palmatilobées ou palmatiséquées par exemple (voire le chapitre 2), si la forme est régulière d'une feuille à l'autre, il est possible d'introduire ces connaissances dans la différenciation. Par exemple, on s'attend à trouver 3 folioles dans une feuille de trèfle. On peut donc introduire l'idée qu'un élément non linéaire contenant 3 extrema locaux est une feuille pour cette plante. Ces extensions n'ont pas été nécessaires pour les plantes de notre étude.

Algorithm 2: Differentiation between apices and leaf blades

Require: Instance segmented similarity graph $G_s = (V, E, P)$ with $\text{card}(P) = p$ and each vertex v associated to an instance $P(v)$. A first classification $C = \{\{\text{leafblades/apices}\}, \{\text{linearorgans}\}\}$ exists such that each instance u in P is associated to a class $\{\text{leafblades/apices}\}$ or $\{\text{linearorgans}\}$. Fiedler vector computed such that $\forall v \in V, f(v)$ is the Fiedler vector value on v .

Ensure: The class $\{\text{leafblades/apices}\}$ is divided into two new classes $\{\text{leafblades}\}$ and $\{\text{apices}\}$.

$\text{dictextremum} :=$ dictionary of p keys corresponding to each instance in P , values initialized to 0

$\text{extrem} : \text{boolean}$

$P_{\text{leafblades/apices}} := \{ \text{instances classified in } \{\text{leafblades/apices}\} \}$

for $i \in V$ **and** $P(i) \in P_{\text{leafblades/apices}}$ **do**

$\text{extrem} = \text{True}$

for each neighbor j of i in V **do**

if $f(j) > f(i)$ or $f(j) < f(i)$ **then**

$\text{extrem} = \text{False}$

end if

end for

if $\text{extrem} = \text{True}$ **then**

$\text{dictextremum}[P(i)] += 1$

end if

end for

for $u \in P$ **do**

if $\text{dictextremum}[u] > 1$ **then**

 Add u to $\{\text{apices}\}$

else

 Add u to $\{\text{leafblades}\}$

end if

end for

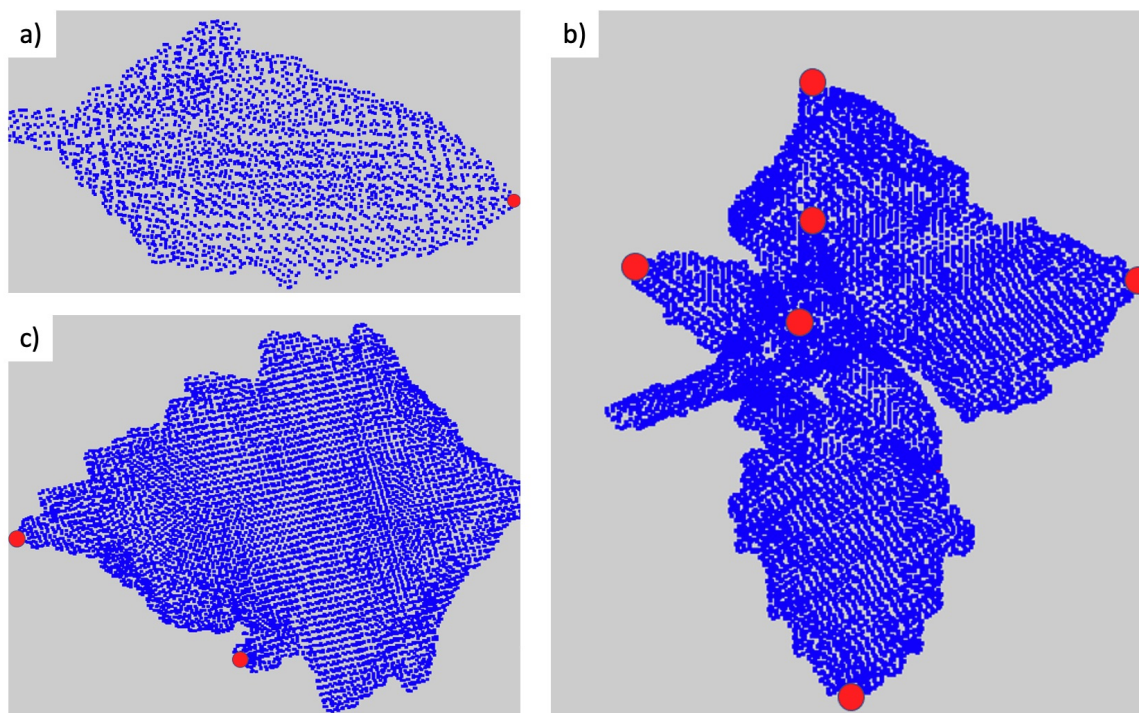


Figure 5.4: Extrema locaux des valeurs du vecteur de Fiedler sur trois organes différents. a) Un limbe b) un apex c) un limbe dentelé.

5.2.3 Charge foliaire : la détermination de la tige

L'algorithme développé pour déterminer la tige principale est entièrement basé sur la structure du graphe quotient et est décrit en pseudo-code dans l'algorithme 3. Il consiste à associer une *charge foliaire* en chaque nœud du graphe quotient. On considère que chaque apex ou limbe fait porter une *charge foliaire* aux nœuds qui permettent de supporter ces organes foliaires. La charge foliaire représente la quantité de limbes ou d'apex chaque organe supporte dans la plante. Par exemple, si la charge foliaire est égale à 1 pour chaque apex ou limbe, la charge d'une branche supportant deux limbes sera égale à 2. Une alternative consisterait à considérer autrement la quantité supportée en prenant en compte le nombre d'extrema du vecteur de Fiedler (section précédente) ou le nombre de points constituant la partie ou encore le volume du limbe ou de l'apex. Dans notre chaîne de traitement, nous avons utilisé la définition la plus simple, avec une charge unitaire définie pour chaque limbe/apex. Les charges foliaires sont calculées de façon récursive, en partant des limbes et des apex. Au début de l'algorithme, tous les nœuds portent une charge de 0, exceptés les limbes et apex qui portent une charge égale à 1. On calcule alors, pour chaque apex et pour chaque limbe, le plus court chemin sur le graphe quotient qui le relie au nœud racine. La charge du nœud ou du limbe sur le plus court chemin est ajouté à la charge de chaque nœud constituant le chemin, comme illustré en figure 5.5. Une fois que toutes les

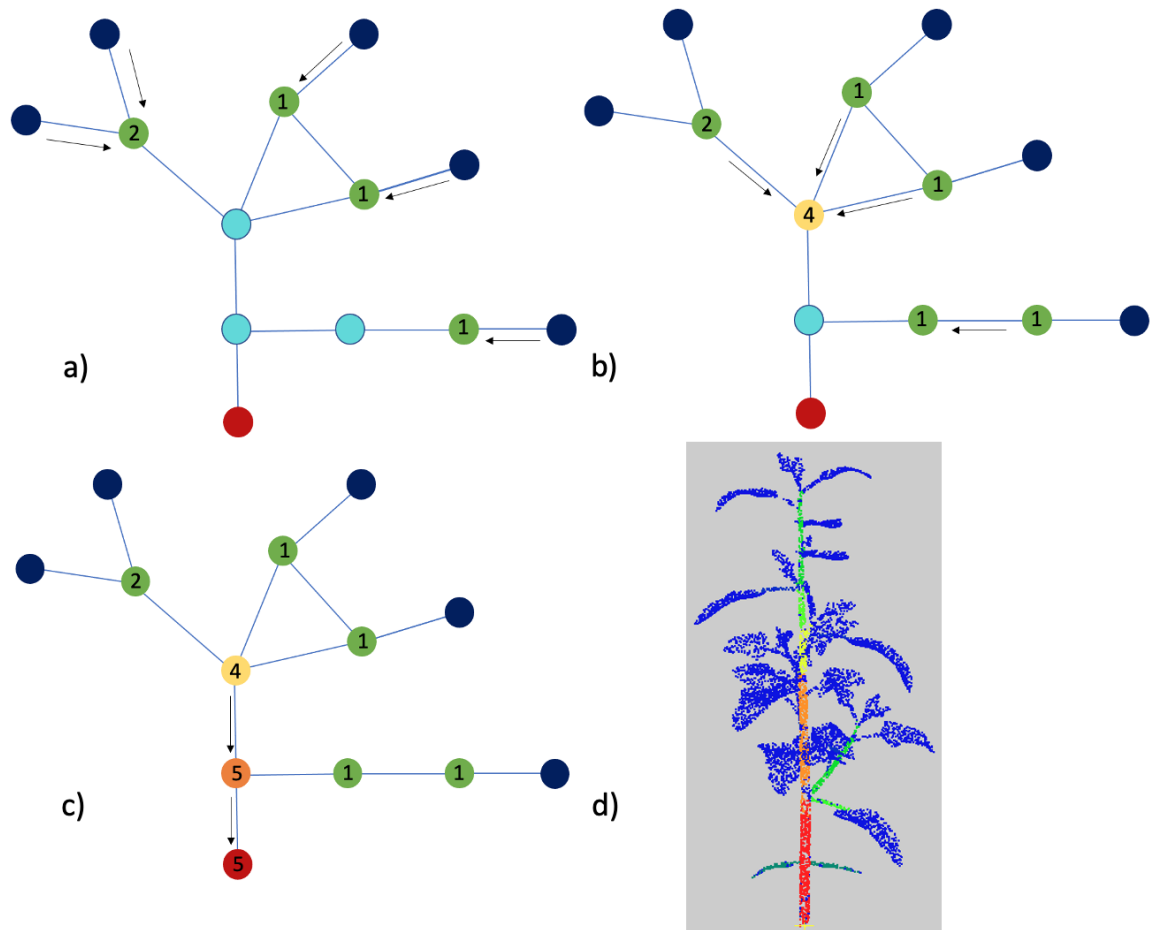


Figure 5.5: Détermination de la tige principale avec le graphe quotient. Le nœud racine est présenté en rouge. Les nœuds représentant les organes linéaires indifférenciés sont en bleu clair au début de l'algorithme. Les nœuds représentant des apex ou des limbes sont en bleu foncé. a)-c) Les plus courts chemins de chaque limbe ou apex jusqu'au nœud racine sont calculés et une charge est ajoutée à chaque nœud pour chacun des chemins qui l'emprunte. d) Le nuage de points présente une segmentation, chaque point a été coloré en fonction de la charge calculée pour son cluster. L'échelle de couleur va du bleu (minimum) vers le rouge (maximum).

charges des nœuds ont été calculées, on définit la tige principale en identifiant le chemin du graphe qui cumule la plus grande charge foliaire à partir de la racine. Pour sélectionner ce chemin de forte charge, nous ne sommes pas toutes les charges foliaires associées aux nœuds du chemin directement, autrement le cumul dépendrait directement du nombre de nœuds présents sur le chemin. Nous ne comptons que les charges foliaires aux nœuds d'embranchements du graphe quotient. Par exemple, en figure 5.5, le chemin associé au limbe le plus bas dans la plante donne les charges suivantes : 1, 1, 5, 5. Le cumul de charge que nous calculons est égal à $1+5=6$ et non pas $1+1+5+5=12$.

Algorithm 3: Main stem detection

Require: Instance segmented similarity graph $G_s = (V, E, P)$ with $card(P) = p$ and each vertex v associated to an instance $P(v)$. A first classification $C = \{\{leafblades\}, \{apices\}, \{linearorgans\}\}$ exists such that each instance u in P is associated to a class in C . The quotient graph $Q_g = (\mathcal{V}, \mathcal{E})$ has been built such that $\mathcal{V} = P$.

Ensure: The creation of a class $\{mainstem\}$ with vertices of Q_g initially classified in $\{linearorgans\}$.

leafload := dictionary of $card(\mathcal{V})$ keys corresponding to every vertex in graph Q_g , values initialized to 0

path := list of $v \in \mathcal{V}$

maxpath := list of $v \in \mathcal{V}$

rootnode := vertex given or automatically computed by the user classified in $\{linearorgans\}$ corresponding to the main stem instance closest to the root

dictofpaths := dictionary of paths computed with the source of path as key (every $v \in \mathcal{V}$ and $v \in \{apices\} \cup \{leafblades\}$)

weight := dictionary of $card(\mathcal{E})$ keys corresponding to every edge in graph Q_g , ponderation of the graph is based on the euclidean distance between mean positions of the instances

for $v \in \mathcal{V}$ and $v \in \{apices\} \cup \{leafblades\}$ **do**

path = *DijkstraShortestPath*(*source* = v , *target* = *rootnode*, *weight*)

dictofpath[v] = *path*

for $u \in path$ **do**

leafload[u] $+$ = 1

end for

end for

previousmaxpathload := 0

for $path \in dictofpath$ **do**

pathload := 0

listloads := list of loads

for $u \in path$ **do**

 Add *leafload*[u] to list *listloads*

end for

 Remove duplicates in list *listloads*

pathload = Sum all values in list *listloads*

if *pathload* > *previousmaxpathload* **then**

maxpath = *path*

previousmaxpathload = *pathload*

end if

end for

for $u \in maxpath$ and $u \in \{linearorgans\}$ **do**

$u \in \{mainstem\}$

end for

Potentiellement, cet algorithme permet également de voir la hiérarchie entre les différents éléments linéaires de la plante. Lorsque la plante devient plus complexe, elle permettrait de distinguer les branches d'ordre 1 (directement reliées à la tige principale), d'ordre 2, etc. Cet algorithme présente des difficultés liées à la qualité de segmentation par instance sous-jacente. Lorsqu'un pétiole ou une branche vient à ne pas être segmenté suffisamment au niveau de son embranchement avec la tige principale, le nœud qui représente le pétiole ou la branche, se retrouve à constituer un nœud de passage systématique entre des parties foliaires plus hautes dans la plante et le nœud racine. La branche ou le pétiole sera incorporé dans la tige principale dans ce cas. Il est donc préférable de disposer d'une sur-segmentation en entrée de cet algorithme plutôt que d'une sous segmentation.

A la fin de cet algorithme, il est possible de fusionner les nœuds du graphe quotient classés en tige principale. Le graphe quotient prend une forme plus simple à analyser. Les nœuds du graphe quotient sont étiquetés limbe, apex, tige principale et organes linéaires.

5.2.4 Plus courts-chemins : l'identification des branches et des pétioles

Une fois que les limbes, les apex et la tige principale ont été identifiés (figure 5.6a)), il reste à détecter les pétioles ainsi que les branches qui se trouvent étiquetés en tant qu'organes linéaires indifférenciés à ce stade. Le pseudo-code pour réaliser cette étape est disponible en 4. Nous mettons en place un système de pondération en fonction des connaissances botaniques sur le placement des organes dont on dispose. Les arêtes connectant des organes qui ne sont pas connectés d'un point de vue botanique reçoivent une pondération infinie (disproportionnellement plus haute que les autres arêtes). Les connexions concernées par cette forte pondération sont les suivantes :

- Limbe-Limbe
- Limbe-Apex
- Apex-Apex
- Limbe-Tige principale
- Apex-Tige principale

Les autres arêtes ont un poids égal à 1. Nous déterminons d'abord la position des branches, nous savons que les apex sont liés à la tige principale par une branche (à l'exception de l'apex de la tige principale). Un plus court chemin de chaque apex

vers la tige principale est calculé dans le graphe quotient. Chaque nœud de ce plus court chemin faisant partie des organes linéaires indifférenciés, est étiqueté en tant que branche (figure 5.6b)).

Le même procédé est appliqué pour déterminer les pétioles, nous savons que qu'un limbe est raccordé à une branche ou à la tige principale grâce à un pétiole. Maintenant qu'une nouvelle classe a été ajoutée, la pondération est complétée. Un nouveau type d'arête reçoit une pondération maximale : Branche-Limbe. Un plus court chemin est calculé entre chaque limbe et la tige principale. Chaque nœud étant encore indifférencié à ce stade et qui se trouve sur un de ces plus courts chemins est étiqueté en tant que pétiole (figure 5.6c)).

Algorithm 4: Identification of branches

Require: Instance segmented similarity graph $G_s = (V, E, P)$ with $card(P) = p$ and each vertex v associated to an instance $P(v)$. A first classification $C = \{\{leafblades\}, \{apices\}, \{mainstem\}, \{linearorgans\}\}$ exists such as each instance u in P is associated to a class in C . The quotient graph $\mathcal{Q}_g = (\mathcal{V}, \mathcal{E})$ has been built such as $\mathcal{V} = P$.

Ensure: The creation of a class $\{branches\}$ with nodes initially classified in $\{linearorgans\}$.

$weight :=$ dictionary of $card(\mathcal{E})$ keys corresponding to every edge in graph \mathcal{Q}_g , values initialized to 0

$path :=$ list of $v \in \mathcal{V}$

$rootnode :=$ vertex in $\{mainstem\}$

for $v_1 \in \mathcal{V}$ and $v_2 \in \mathcal{V} \setminus \{v_1\}$ such that $(v_1, v_2) \in \mathcal{E}$ **do**

if $(v_1 \in \{leafblades\}$ and $v_2 \in \{leafblades\})$ OR $(v_1 \in \{leafblades\}$ and $v_2 \in \{apices\})$ OR $(v_1 \in \{apices\}$ and $v_2 \in \{apices\})$ OR $(v_1 \in \{leafblades\}$ and $v_2 \in \{mainstem\})$ OR $(v_1 \in \{apices\}$ and $v_2 \in \{mainstem\})$ **then**

$weight[(v_1, v_2)] = max(integer)$

else

$weight[(v_1, v_2)] = 1$

end if

end for

for $v \in \mathcal{V}$ and $v \in \{apices\}$ **do**

$path = DijkstraShortestPath(source = v, target = rootnode, weight)$

for $u \in path$ **do**

if $u \in \{linearorgans\}$ **then**

$u \in \{branches\}$

end if

end for

end for

Les nœuds qui n'ont pas été classés dans un groupe d'organe, et sont toujours dans la classe linéaire indifférenciée sont fusionnées avec le nœud avec lequel ils sont

le plus connectés (nombres de connexions entre les nœuds du graphe de similarité, reporté au niveau du graphe quotient, connexions inter classes voir section 5.1.2) (figure 5.6d)). D'après nos observations, les parties fusionnées sont le plus souvent composées de peu de points (moins de 100). Il s'agit des parties introduites par l'utilisation d'un critère de variance sans notion topologique décrit dans la section 4.3.3. Toutes les arêtes qui sont nécessaires à un plus court chemin entre les limbes ou les apex jusqu'à la tige principale sont conservées dans le graphe quotient. Les arêtes qui n'ont pas été utilisées dans un plus court chemin dans les étapes précédentes sont supprimées (figure 5.6e)). Cette étape permet de supprimer les cycles et de favoriser les connexions qui ont un sens d'un point de vue botanique. Nous ne conservons ainsi pas toutes les connexions qui existaient d'un point de vue géométrique.

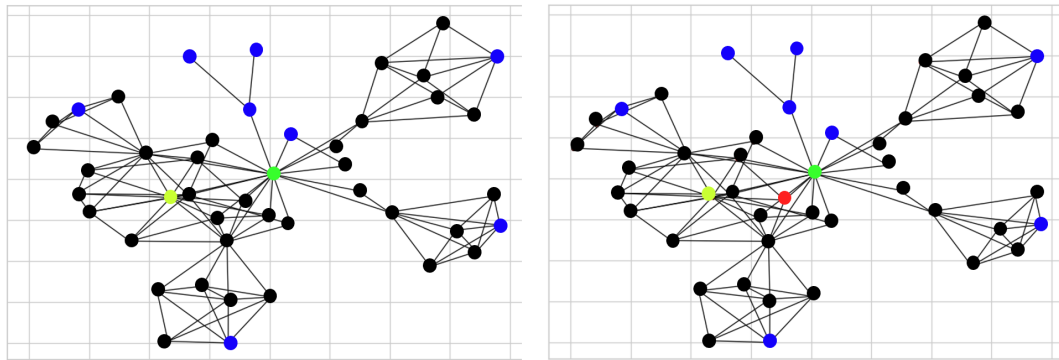
A l'issue de cette étape, on obtient un arbre quotient sémantique. Tous les nœuds ont une étiquette indiquant quel organe de la plante ils représentent. L'arbre a bien une racine : le nœud représentant la tige principale. Les cycles ont été supprimés lors de la suppression des arêtes non nécessaires pour relier les apex ou limbes à la tige principale. Il est encore possible d'effectuer une étape de fusion des nœuds qui seraient adjacents et représenteraient le même organe (figure 5.6f))

5.2.5 Conformité du graphe quotient : amélioration de la segmentation par instance

A ce stade, nous disposons d'une segmentation par instance et d'une segmentation sémantique, ainsi que d'un arbre sémantique. Cette partie consiste à exploiter l'arbre sémantique pour contrôler la qualité et la vraisemblance botanique de la segmentation. Toutes les arêtes restantes ayant un poids infini représentent des anomalies botaniques, des organes qui sont connectés alors qu'ils ne devraient pas l'être. On distingue plusieurs cas de figures que l'on peut détecter :

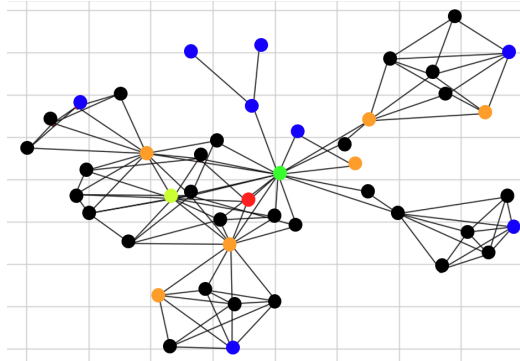
- 1. Un apex ou un limbe est directement connecté à la tige principale dans le graphe quotient.
- 2. Un limbe est connecté à une branche.
- 3. Un nœud correspondant à un organe linéaire (branche ou pétiole) est connecté à plusieurs apex ou limbes.

Dans tous ces cas, un organe est manquant ou n'a pas été correctement identifié. Il est nécessaire d'affiner la segmentation pour corriger ces erreurs. Dans les cas 1 et 2, un pétiole ou une branche n'ont pas été segmentés et sont fusionnés avec le limbe ou l'apex mal connecté. On décide donc de resegmenter le limbe ou l'apex on utilisant

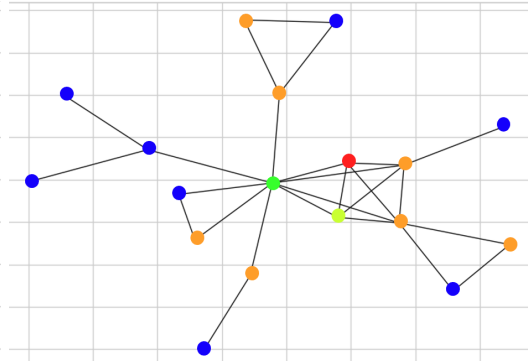


(a) Identification de la tige principale

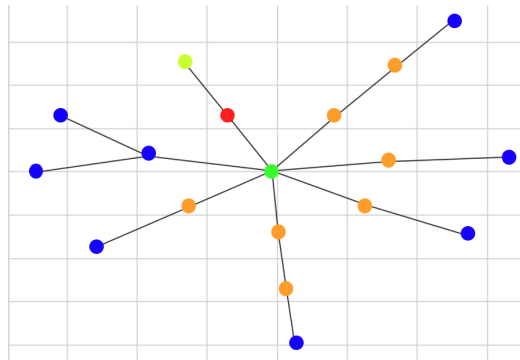
(b) Identification des branches



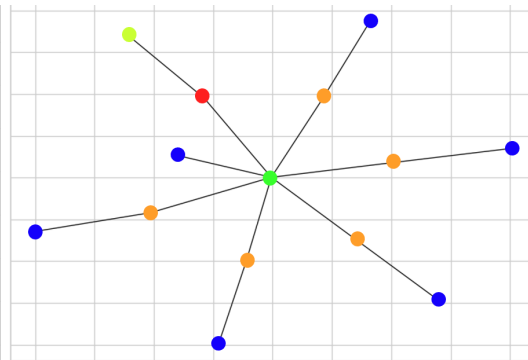
(c) Identification des pétioles



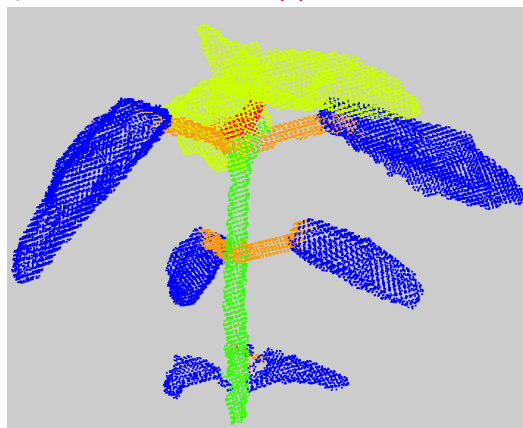
(d) Fusion des nœuds indifférenciés



(e) Suppression des cycles



(f) Fusion de nœuds



(g) Nuage de point correspondant avec segmentation sémantique

Figure 5.6: Évolution du graphe quotient vers un arbre sémantique au fur et à mesure des algorithmes. Chaque couleur correspond à un organe, en bleu les limbes, jaune les apex, orange les pétioles, rouge les branches et vert la tige principale.

la norme du gradient du vecteur de Fiedler combiné à l'algorithme des K-means avec la méthode du coude. Lorsque l'on se trouve dans le troisième cas de figure, il est nécessaire de resegmenter l'organe linéaire en réutilisant les directions du gradient du vecteur de Fiedler avec les k -means et la méthode du coude. Les nouvelles parties sont étiquetées en deux catégories : linéaire ou limbe/apex.

Étant donné les modifications qui ont été apportées à la structure du graphe, les conclusions sémantiques faites dans les étapes précédentes et dépendantes fortement de cette structure sont à reprendre. Potentiellement, il serait possible d'effectuer ce processus d'amélioration tant que l'on trouve des erreurs. Nous avons effectué une seule itération afin notamment de résoudre les problèmes de sous-segmentation d'éléments faussement considérés comme apex. Un apex et la correction apportée sur cet apex y sont présentés en figure 5.7 à titre d'exemple. Cet apex a été détecté comme une anomalie car il était directement rattaché à la tige principale directement. Une fois détecté, il est segmenté à nouveau via l'utilisation de la norme du gradient du vecteur de Fiedler et de la méthode du coude. La partie de norme la plus petite est classée en apex/limbe et les autres parties en classe linéaire indifférenciée. Nous pouvons ainsi relancer les algorithmes de détermination des classes sémantiques présentés précédemment : différenciation limbe/apex, identification des branches et des pétioles. Dans notre exemple, sont alors identifiés un apex, un limbe et une branche. On remarque que la branche n'est pas suffisamment segmentée, elle devrait encore être divisée puis une branche et un pétiole devraient être identifiés. On note également l'effet de l'algorithme sur un cotylédon (entouré en noir), l'algorithme ayant identifié le cotylédon comme un limbe, a cherché à le corriger en y cherchant un pétiole. Dans ce cas, le résultat a été dégradé puisque nous n'avons pas pris en compte la classe des cotylédons et ses particularités dans notre traitement.

Nous verrons dans le chapitre 6 que l'issue de cette étape dépend du jeu de données et du type d'erreurs rencontrées. Un travail futur sur cette partie pourrait consister en l'établissement d'un seuil de convergence pour pouvoir réitérer la boucle de correction. Cela pourrait passer par un score de vraisemblance botanique.

5.3 Conclusion

Nous avons présenté dans ce chapitre la construction du graphe quotient. Ce graphe permet de ne plus considérer le point du nuage de point mais des ensembles de points déjà regroupés dans une segmentation par instance. Nous effectuons donc un changement d'échelle. Il nous est possible de faire remonter des informations qui

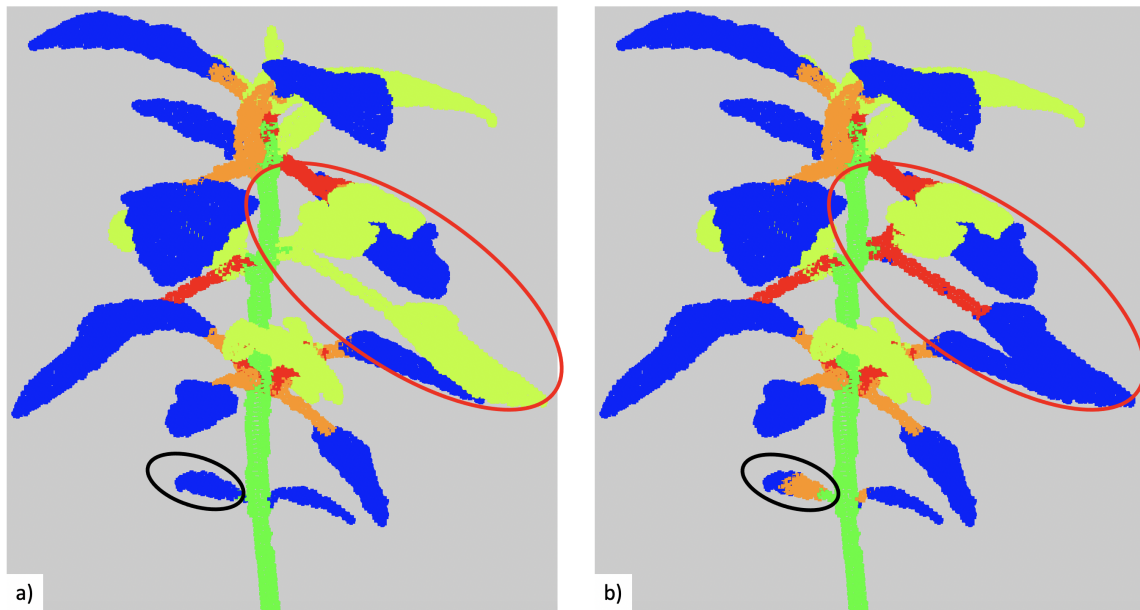


Figure 5.7: Changements apportés par l'étape d'amélioration de la segmentation sur un chénopode réel. La segmentation présentée est une segmentation sémantique, chaque point du nuage a la couleur de la classe à laquelle il appartient. Les limbes sont en bleue, les apex en jaune, la tige principale en vert, les branches en rouge et les pétioles en orange. Une erreur de sous-segmentation d'un apex a été identifiée et entourée en rouge. Un cotylédon est entouré en noir. a) Segmentation sémantique sans étape d'amélioration b) Segmentation sémantique après étape d'amélioration.

étaient présentes au niveau du graphe de similarité comme des attributs du graphe quotient. Nous utilisons cette caractéristique dans l'algorithme de division/fusion de nœuds du graphe quotient ce qui nous permet d'exploiter les directions du GDM du vecteur de Fiedler et finaliser la segmentation par instance à partir d'attributs géométriques entamée dans le chapitre précédent. Les étapes de classifications des nœuds du graphe quotient et du graphe de similarité démarrent après cette étape. Nous effectuons d'abord une différenciation des limbes et des apex via un calcul de maximums locaux. Puis, nous nous appuyons sur l'idée que la tige principale est l'organe qui soutient toutes les feuilles dans l'architecture de la plante pour déterminer les nœuds du graphe quotient qui en font partie. Nous parcourons ensuite le graphe quotient pour identifier les branches et les pétioles en démarrant des limbes ou des apex. Une fois les nœuds correctement classifiés, il est possible d'améliorer la topologie du graphe quotient, en supprimant les cycles et les arêtes aberrantes botaniquement jusqu'à obtenir un arbre sémantique. Nous introduisons également une étape de correction à partir de la détection d'anomalies à l'échelle de l'arbre sémantique. Cette étape supplémentaire semble introduire une amélioration visuelle sur les nuages de

points. Néanmoins, il est nécessaire d'évaluer les deux segmentations obtenues de façon quantitative et c'est ce à quoi est dédié le chapitre suivant.

Chaîne de traitement et évaluation

Nous avons introduit dans les chapitres précédents un certain nombre d’algorithmes pour la segmentation par instance et sémantique d’un nuage de points 3D de plantes. Nous avons proposé deux attributs pour discriminer de façon géométrique les organes, ainsi que des algorithmes pour exploiter ces attributs. Nous avons également montré les limites de cette approche purement géométrique et proposons une nouvelle échelle de traitement avec le graphe quotient. A cette échelle, nous avons pu travailler différemment et également introduire des connaissances botaniques.

Dans ce chapitre, nous proposons une chaîne de traitement articulant ces algorithmes pour obtenir une segmentation par instance et une classification des instances. En effet, la construction du graphe quotient se fait sur la création d’instances au préalable. Tous les algorithmes ne sont pas interchangeables dans le traitement. Par conséquent, beaucoup d’essais et de tests ont été réalisés afin d’obtenir une chaîne de traitement cohérente et efficace. Dans la suite de ce chapitre, nous évaluons les résultats obtenus avec cette chaîne de traitement, et ce sur trois jeux de données différents. La chaîne de traitement propose en sortie plusieurs objets : un nuage de points segmenté par instance, classifié, mais également un arbre sémantique. Nous allons discuter et quantifier quand cela est possible la qualité de ces objets dans ce chapitre.

6.1 Chaîne de traitement

L’ensemble des algorithmes présentés dans les sections précédentes ont été assemblés dans le but d’obtenir une segmentation sémantique et par instance de nuages de points de plantes. On retrouve en figure 6.1, sur la première ligne, un grand nombre des étapes décrites dans le chapitre 4. On construit la structure du graphe de similarité G à partir du nuage de points en entrée. Il est alors possible de calculer la signature spectrale de ce graphe et d’en extraire le vecteur de Fiedler. Nous calculons une forme de pseudo gradient du vecteur de Fiedler, le GDM de Fiedler qui nous permet de disposer de deux attributs, disponibles en chaque point du nuage : la norme et les directions de ce champ de vecteurs.

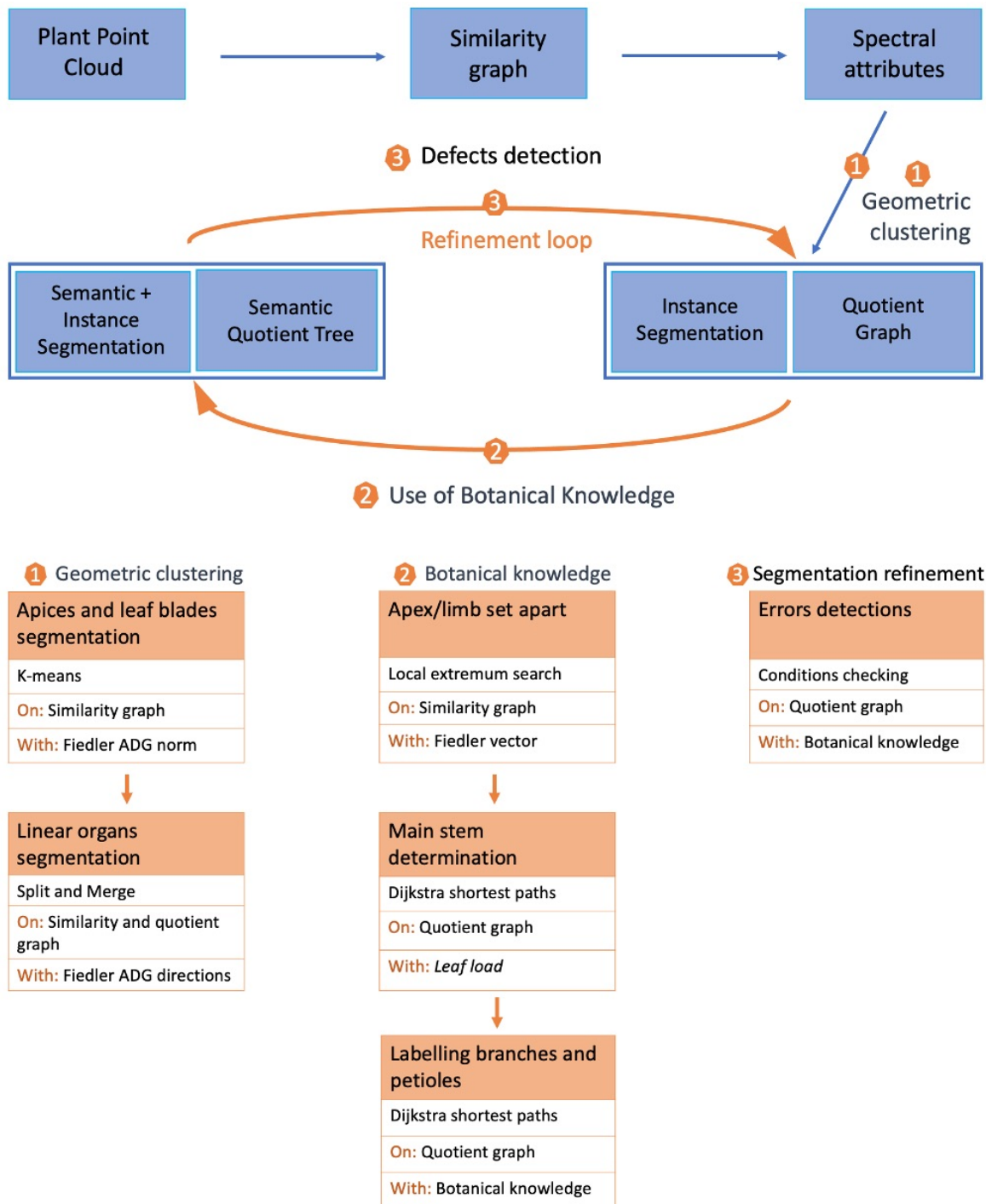


Figure 6.1: Chaîne de traitement mise en place pour la segmentation de nuages de points de plantes.

L'étape 1) consiste alors à exploiter ces attributs pour obtenir une segmentation par instance. Dans un premier temps, on applique l'algorithme des K-means sur la norme du GDM du vecteur de Fiedler avec $K=4$ (section 4.3.1). Cela permet d'isoler les limbes et apex des éléments linéaires de la plante dans la partie de plus petite norme en moyenne. Grâce à un algorithme de croissance de région (section 4.3.2), on obtient, en

plus de cette grossière segmentation sémantique, une segmentation par instance. Afin d'exploiter les directions du GDM du vecteur de Fiedler, nous construisons un nouvel objet, le graphe quotient (section 5.1.1). Cela nous permet d'appliquer l'algorithme de division/fusion présenté en section 5.2.1. A l'issue de cette étape, on dispose donc d'une segmentation par instance issue de connaissances géométriques. On dispose également d'une classification sémantique grossière, les limbes et les apex d'un côté, les parties linéaires (pétioles, tige principale, branches) de l'autre. On dispose également d'un graphe quotient permettant d'effectuer des traitements à une autre échelle, celle des parties déjà générées.

La construction de ce graphe quotient permet alors dans l'étape 2) d'utiliser des connaissances botaniques élémentaires afin d'obtenir les classes sémantiques des éléments déjà segmentés par instance. Après avoir différencié les apex et les limbes en nous appuyant sur le vecteur de Fiedler, nous nous appuyons sur la nature de l'objet pour affiner les classes sémantiques. Les nœuds appartenant à la tige principale sont déterminés grâce à la nature hiérarchique de la plante et à l'application d'un algorithme de Dijkstra de recherche de plus court chemin sur le graphe quotient (section 5.2.3). De façon similaire, nous prenons en compte les relations entre les organes pour distinguer les pétioles des branches (section 5.2.4). En parallèle de l'obtention de ces classes sémantiques, le graphe quotient se précise et devient même un arbre car nous avons éliminé les cycles et le nœud racine est donné. Cet arbre est dit sémantique car chaque nœud est identifié comme un organe de la plante. Enfin, dans l'étape 3), il est possible de contrôler l'arbre sémantique en regardant si toutes les connections entre les organes font sens d'un point de vue botanique. Il est également possible de détecter les erreurs potentielles et l'on peut alors automatiquement les corriger en reprenant l'étape 1) et l'ensemble de la boucle (section 5.2.5).

Le temps de calcul de cette chaîne de traitement dépend majoritairement du nombre de points des nuages traités, comme le présente la figure 6.2. Néanmoins, la complexité de la plante à l'étude, qui peut ainsi induire un nombre plus important de parties à traiter, peut également avoir un impact non négligeable mais plus difficile à quantifier sur le temps de calcul.

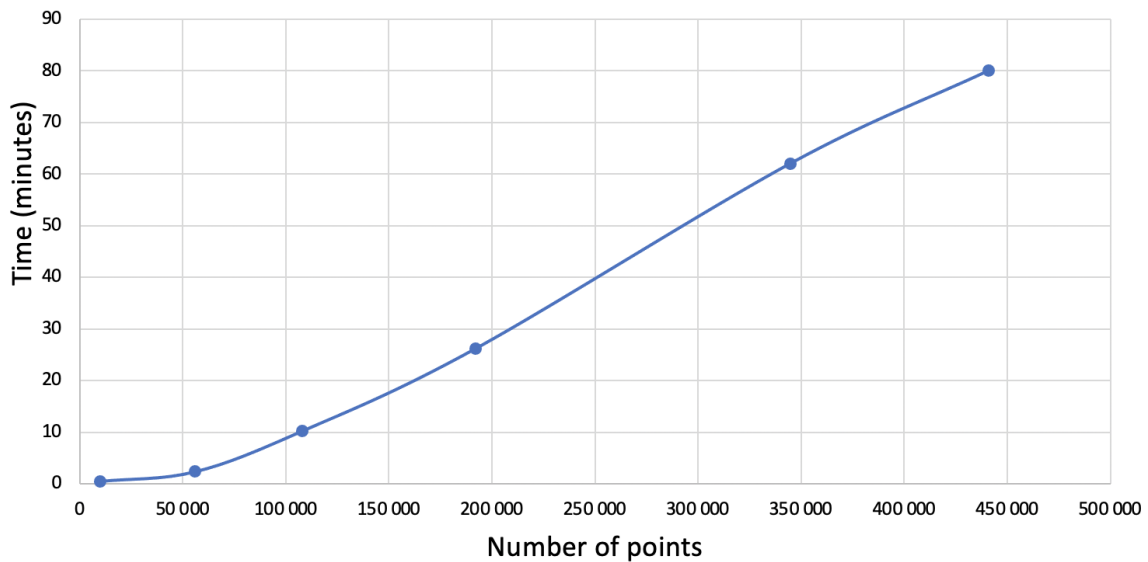


Figure 6.2: Temps de calcul de l'ensemble de la chaîne de traitement en fonction du nombre de points présents dans le nuage. La première mesure a été effectuée sur un chénopode virtuel, les deux suivantes sur des chénopodes réels et les trois dernières sur des plants de tomates issus de Pheno4D. Les plantes ont été sélectionnées de façon à présenter des difficultés et un nombre de parties à atteindre relativement égales.

6.2 Jeux de données de validation

L'évaluation a été effectuée sur trois jeux de données différents :

- Un jeu de données constitué de nuages de points de chénopodes réels issus du projet ROMI décrit en section 2.3.1, illustré en figure 6.3 a) et d).
- Un jeu de données constitué de nuages de points de chénopodes virtuels issus du projet ROMI, illustré en figure 6.3 b) et e).
- Un jeu de données constitué de plants de tomates réels issus du projet Pheno4D de SCHUNCK et al., 2021, illustré en figure 6.3 c) f).

6.2.1 Jeu de données des chénopodes réels

Le chénopode est considéré comme une mauvaise herbe et a une croissance rapide et monopodiale. Celle-ci crée une tige principale bien définie qui est plus longue et dispose d'un diamètre plus important que les branches latérales. La croissance de la tige principale est plus rapide que celles des axes secondaires. Les feuilles sont simples, relativement petites et de forme triangulaire. La plante produit des inflorescences après quelques semaines seulement. Dans notre étude, les plantes sont acquises avant qu'elles n'atteignent le stade d'inflorescence.



Figure 6.3: Jeux de données utilisés pour l'évaluation du pipeline. Photographies ou captures d'écran des objets de a) à c) : a) chénopode réel, b) chénopode virtuel, c) plant de tomates (photo d'illustration, indépendante du jeu de données de SCHUNCK et al., 2021). Les nuages de points correspondants sont présentés de d) à f). On notera que d) et e) ont les mêmes angles de vues et représentent exactement les objets présentés en a) et b).

Des graines de chénopodes issues de boutiques de graines en ligne ont été utilisées. Les chénopodes ont été cultivés dans des chambres de cultures sur un sol argilo-tourbeux (concentration d'argile : 60kg.m^{-3}), arrosés avec un engrais (18 - 10 - 18N - P - K) et éclairés avec un système LED (Valoya©, spectre NS12, $150\text{mol.m}^{-2}.s^{-1}$) avec une alternance jour/nuit composée de 16 heures de jour et 8h de nuit. La température et l'humidité étaient de 22 degrés Celsius et 60 % d'humidité le jour et de 18 degrés Celsius et 70 % d'humidité la nuit. 8-10 jours après germination des graines, les plants ont été transférés dans des pots individuels. Ils ont été imagés entre 6 à 12 semaines après germination.

72 photographies sont prises afin d'imager un plant de chénopode. Les photos sont prises régulièrement autour d'un axe circulaire autour de la plante. Le ROMI Plant Imager a été utilisé et a déjà été décrit en Section (2.3.1).

Afin de construire notre segmentation de référence, nous avons sélectionné 5 nuages de points dans ces données. Ces 5 nuages de points ont été annotés manuellement de façon à disposer d'une segmentation sémantique et par instance de référence. Nous avons utilisé le logiciel CloudCompare REF pour isoler chaque organe de l'ensemble du nuage de points. Nous attribuons une étiquette à l'ensemble des points qui constitue cet organe. Nous avons fait le choix d'associer à chaque organe un nombre qui nous permettrait par la suite de récupérer la classe sémantique et l'instance de la partie. Les apex sont numérotés dans une plage entre 300 et 399 par exemple, les limbes entre 400 et 499, les pétioles entre 200 et 299, les branches entre 100 et 199 et la tige principale est numérotée 0. Enfin, lorsque chaque organe a été séparé et étiqueté correctement, nous les fusionnons tous pour obtenir le nuage de points de référence.

6.2.2 Jeu de données des chénopodes virtuels

Des nuages de points virtuels ont été générés à partir de plantes virtuelles pour pouvoir également évaluer l'approche. Les nuages de points ont été générés selon la méthode décrite dans Chaudhury et al., 2020. La croissance des chénopodes a été modélisée en utilisant les L-systèmes et le logiciel L-Py (BOUDON et al., 2012). La surface des modèles virtuels est ensuite utilisée pour obtenir un nuage de points étiqueté. Par construction du modèle L-Py, les plantes ainsi obtenues sont étiquetées en trois classes : pétiole, tige et limbe. Le modèle ne contient pas la classe apex et ne distingue pas dans la classe tige, la tige principale des branches secondaires.

Par conséquent, le résultat de la chaîne de traitement a dû être adapté pour permettre une comparaison des résultats. Les apex étant des agglomérats de petites feuilles non complètement formées, ils ont été considérés comme des limbes lors de la comparaison. De la même manière, les classes branches et tige principale ont été fusionnées en une unique classe tige.

Au total, 24 nuages de points de chénopodes ont été générés. Chaque nuage était constitué de 10000 points. La segmentation par la chaîne de traitement de ces 24 nuages a été effectuée en 12 minutes sur un ordinateur disposant d'un processeur 2,6 GHz Intel Core i7 et de 32 Go de mémoire.

6.2.3 Tomates réelles issues de Phéno4D

Les plants de tomates ont une croissance sympodiale, contrairement au chénopode. Ils produisent des feuilles composées et présentent souvent des architectures plus complexes. Afin de tester notre chaîne de traitement sur une espèce de plante différente du chénopode, nous avons utilisé des nuages de points de tomates issus du jeu de données Pheno4D produit par SCHUNCK et al., 2021. Ce jeu de données contient des nuages de points de 7 plants de tomates imagées quotidiennement pendant trois semaines, dès la sortie de terre des plantes. Il contient au total 140 nuages de points de tomates dont 77 classifiés à la main. On notera que ce jeu de données contient également des acquisitions segmentées de 7 plants de maïs que nous avons laissés de côté pour notre étude. Les plantes ont été cultivées en pot dans une station de culture sous serre. Les nuages de points ont été acquis avec un scanner laser avec une précision inférieure au millimètre. Nous avons extrait de ce jeu de données 14 nuages de points correspondant à deux individus différents observés à différents moments de leur croissance. Parmi les moments de croissance disponibles, nous avons laissé de côté les stades de plantules car ils étaient hors du cadre de notre étude. Nous avons donc conservé 7 moments de croissances pour chacune des deux plantes. Les nuages de points sont fournis avec une segmentation de référence. Cette segmentation consiste en une classe tige, contenant toutes les parties linéaires de la plante (pétiole, tige principale, branches), une classe limbe et une classe sol (le pot de la plante). On notera que la segmentation de référence distingue également chaque limbe les uns des autres. On dispose donc d'une segmentation par instance de référence pour cette classe. Les nuages de points de ce jeu de données sont très denses et peuvent dépasser 4 millions de points pour une plante. Afin d'avancer plus rapidement dans notre analyse, nous avons re-échantillonné les nuages en augmentant la distance minimale entre deux points. Nous avons ainsi obtenu des nuages comprenant entre 50 000 et 500 000 points.

Le résultat de la chaîne de traitement a dû être adapté pour être comparé aux modèles de références issus des chénopodes virtuels. Tout d'abord, la classe sol du jeu de données n'a pas été conservée, les points appartenant à cette classe ont été retirés du modèle. L'ensemble des parties linéaires (pétioles, tige principale, branches) ont été fusionnées en une classe, les classes limbes et apex ont également été fusionnées en une même classe. On notera que nous n'avons utilisé ce jeu de données que pour faire une évaluation sémantique et non pas par instance des limbes. Cela est dû au fait que notre chaîne de traitement ne dispose pas d'une instance sur les limbes de ce type de plantes. En effet, l'algorithme utilisé pour obtenir les instances entre les limbes est un algorithme de croissance par région (section 4.3.2). Par conséquent, sur

une plante telle que la tomate où les limbes se touchent, la chaîne de traitement ne permettra pas d'en distinguer les instances.

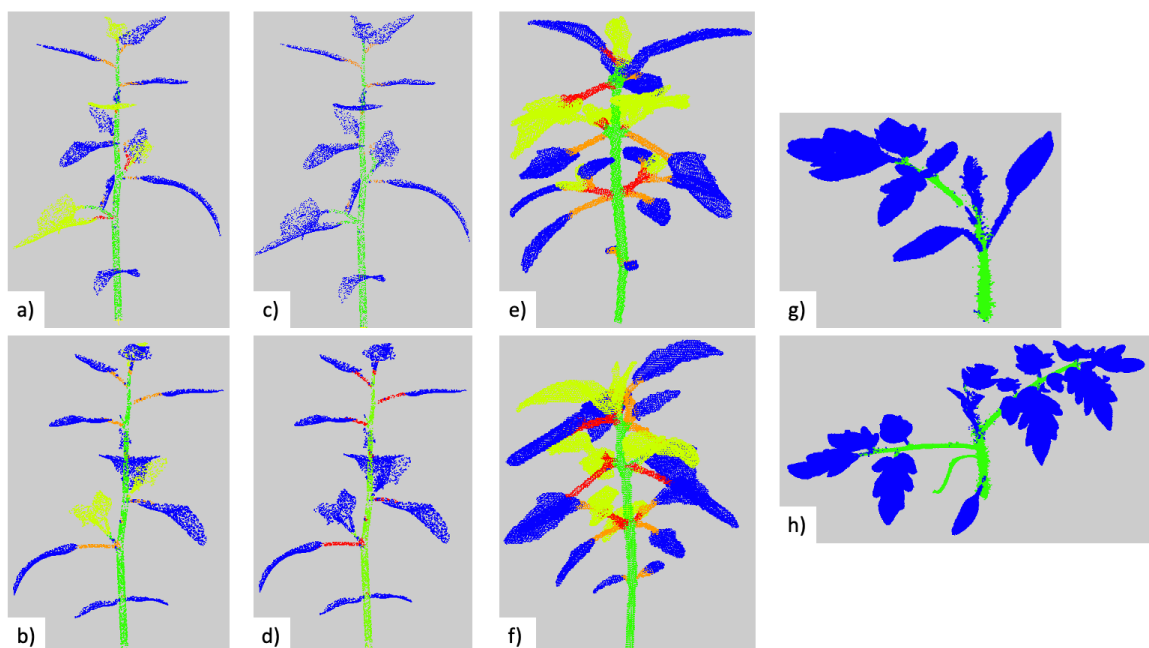


Figure 6.4: Résultats visuels de la segmentation sémantique réalisée par la chaîne de traitement. Les points sont colorés en fonction de la classe sémantique qui leur est attribuée, bleu pour les limbes, vert pour la tige principale, orange pour les pétioles, rouge pour les branches et jaune pour les apex. a-b) Segmentation sémantique de deux chénopodes virtuels différents. c-d) Même segmentations sémantiques que les deux chénopodes présentés en a-b) mais les classes apex ont été fusionnées avec les limbes. e-f) Segmentation sémantique de deux chénopodes réels. g-h) Segmentation sémantique de deux plants de tomate réels.

6.3 Résultats

6.3.1 Métriques d'évaluation

Afin d'évaluer la qualité de la segmentation, une approche consiste à comparer le résultat d'une segmentation issue de la chaîne de traitement présenté, avec une segmentation témoin, experte, de vérité terrain. On peut compter le nombre TP_c de points du nuage classés dans la classe c qui portent la même étiquette dans le résultat obtenu, autrement dit les vrais positifs de la classe c . On note FN_c , les faux négatifs de la classe c , c'est-à-dire les points qui n'ont pas été classés en c dans le résultat mais qui appartenaient bien à c . On note FP_c les faux positifs de la classe c , c'est-à-dire les points qui ont été classés en c dans le résultat mais appartenaient à une autre classe. A partir de ces valeurs, il est possible de calculer différentes statistiques classiques pour

la classe c : le rappel Re_c , la précision Pr_c , IoU_c (intersection sur union), ou encore le F_1 -score. Elles sont définies de la manière suivante :

$$Re_c = \frac{TP_c}{TP_c + FN_c} \quad (6.1)$$

$$Pr_c = \frac{TP_c}{TP_c + FP_c} \quad (6.2)$$

$$IoU_c = \frac{TP_c}{TP_c + FN_c + FP_c} \quad (6.3)$$

$$F1_c = \frac{TP_c}{TP_c + 0.5 * (FP_c + FN_c)} \quad (6.4)$$

Ces statistiques sont le plus souvent utilisées pour évaluer les résultats d'une segmentation sémantique. Néanmoins, elles peuvent être utilisées pour évaluer une segmentation par instance à la condition que l'on dispose d'une table de correspondance entre chacune des parties, autrement dit, que l'on puisse automatiquement associer chaque partie du résultat segmenté par la chaîne de traitement à la segmentation par instance de référence. Cela pose un certain nombre de difficultés même si l'on arrive à faire cette correspondance. Par exemple, si une instance de la référence a été découpée en deux instances dans le résultat et que l'on a pris le plus gros des deux (en terme de nombre de points), que fait-on du deuxième ?

Lorsque l'on ne dispose pas des correspondances ou que l'on souhaite disposer de statistiques sur la segmentation par instance qui puisse gérer et rendre compte de la possibilité d'un nombre différent d'instances entre la référence et le résultat, on peut utiliser un certain nombre de scores. Ces scores sont basés sur l'observation des classes de points deux à deux. L'indice de Rand est un bon exemple de ce type de scores. On considère un point $p1$ et un point $p2$ du nuage, si les deux points sont dans la même partie C_1 dans la segmentation résultat et dans la segmentation référence, on peut compter ce résultat comme un vrai positif TP . Si les deux points sont dans des parties différentes dans la segmentation référence comme dans la segmentation résultat, il est possible de considérer ce résultat comme un vrai négatif TN . Dans le cas où les deux points sont considérés dans la même partie dans la segmentation de référence mais dans des parties différentes dans la segmentation résultat, il s'agit d'un

faux négatif FN . A l'inverse de la situation précédente, il s'agira d'un faux positif FP . A partir de ces cas, il est possible de calculer l'indice de Rand :

$$RI = \frac{TP + TN}{TP + FP + FN + TN} \quad (6.5)$$

L'indice de Rand est toujours compris entre 0 et 1. Lorsqu'il est proche de 0, cela signifie que la segmentation résultat ne coïncide sur aucune paire d'éléments avec la segmentation de référence. Lorsqu'il est proche de 1, cela signifie que les deux segmentations coïncident. Si l'on comprend que la valeur 1 est associée à une correspondance complète entre les deux parties, atteindre la valeur 0 ne paraît pas aussi simple. Même si l'on obtenait une segmentation résultat mettant tous les points dans leur propre partie, on obtiendrait quand même un certain nombre de vrais négatifs TN . L'indice de Rand existe donc également en version "ajustée" de façon à attribuer la valeur 0 si les deux parties sont statistiquement indépendantes (WARRENS et HOEF, 2022). Par la suite on utilisera également l'indice "AMI" (Adjusted Mutual Information), plus adapté dans le cas où il existe de fortes disparités de tailles entre les parties (ROSENBERG et HIRSCHBERG, 2007) et ajusté pour les mêmes raisons que l'indice de Rand. Enfin, on utilisera encore deux indices supplémentaires. L'indice d'homogénéité (homogeneity) est un indice qui est d'autant plus haut que chacune des parties de la segmentation résultat ne contient que des points issus de la même partie de la segmentation de référence et très peu de points issus d'autres parties. De façon complémentaire, l'indice d'exhaustivité (completeness) est d'autant plus haut que tous les éléments qui devraient se trouver dans la même partie sont bien rassemblés dans la segmentation résultat.

6.3.2 Evaluation de la segmentation sémantique

Sur les chénopodes virtuels

Nous avons tout d'abord testé notre approche sur le jeu de données des chénopodes virtuels. La chaîne de traitement a été exécutée avec et sans la phase d'amélioration de la segmentation (section 5.2.5). La chaîne de traitement permet d'obtenir en sortie une classification sémantique en 5 classes : limbes, apex, pétioles, branches, tige principale. Néanmoins, le modèle virtuel ne nous permettait d'avoir une segmentation sémantique qu'avec 3 classes de référence : limbes, pétioles, tige (branches + tige principale). Nous avons donc fusionné certaines classes de notre algorithme pour nous conformer aux classes de la référence (branches avec tige principale et apex avec

limbes). Nous avons ensuite appliqué les métriques décrites précédemment pour la segmentation sémantique, les résultats sont présentés en table 6.1.

On peut noter que les limbes sont globalement bien segmentés et présentent des valeurs de rappel et de précision hautes, ainsi qu'un IoU de 90%. La procédure d'amélioration n'a pas modifié de façon significative les valeurs du tableau. Ces résultats sont cohérents avec le macro F_1 -score obtenu, de 91% et de 88%, respectivement sans et avec l'étape d'amélioration. Les tiges sont également bien segmentées de façon générale même si les résultats sont légèrement plus bas. En revanche, la classe des pétioles dispose de scores de rappel et de précision très contrastés. La précision (68 %) est bien plus élevée que le rappel (30 %). Beaucoup de points faisant partie des pétioles n'ont donc pas été détectés, néanmoins, le peu de points identifiés comme pétiole étaient majoritairement bien des points appartenant à des pétioles. Dans l'ensemble les pétioles sont très mal détectés par l'algorithme avec un IoU très faible.

Classe	Sans amélioration			Avec amélioration		
	Re	Pr	IoU	Re	Pr	IoU
Limbe	95	92	90	94	94	88
Tige	88	91	81	91	87	80
Pétiole	30	68	26	38	51	27

Table 6.1: Résultats de la segmentation sémantique (en %) sur le jeu de données des chénopodes virtuels, avec et sans l'étape d'amélioration (section 5.2.5)

La mauvaise détection des pétioles est due à quatre raisons principales. Premièrement, les pétioles sont souvent de petits organes. La classe pétiole contient donc moins de points que la classe des limbes ou des tiges. Étant donné que les frontières entre les organes ne sont pas toujours précisément déterminées dans les segmentations manuelles ou automatiques, il est possible d'observer des erreurs de classification arbitraires plus importantes aux frontières (figure 6.5a) et b). Dans le cas des pétioles, ces erreurs, trouvées aux extrémités d'organes contenant peu de points, va avoir un impact fort sur la qualité globale de la définition de l'organe. Deuxièmement, il peut arriver qu'un limbe soit bien délimité mais détecté comme un apex dans notre algorithme, le pétiole associé est alors classifié en branche dans notre algorithme et par conséquent sera classifié en tige et non détecté (section 5.2.2). Troisièmement, il peut arriver que l'algorithme de détermination de la tige ait englobé un pétiole dans la tige principale (figure 5.4c)). Sur ce jeu de données, cela peut s'expliquer par une densité de points très faible pour la définition des organes linéaires, la séparation tige/pétiole est donc approximative et provoque une erreur décrite en section 5.2.3. Cette erreur n'est pas correctement corrigée par l'étape d'amélioration car notre algorithme cherche l'organe manquant, le pétiole dans la partie de classe limbe et non dans la tige (figure 5.4d)).

Enfin, il arrive que le limbe soit fusionné avec son pétiole comme en figure 6.5 e). Néanmoins, l'étape d'amélioration tend à corriger cette erreur, voir la figure 6.5 f).

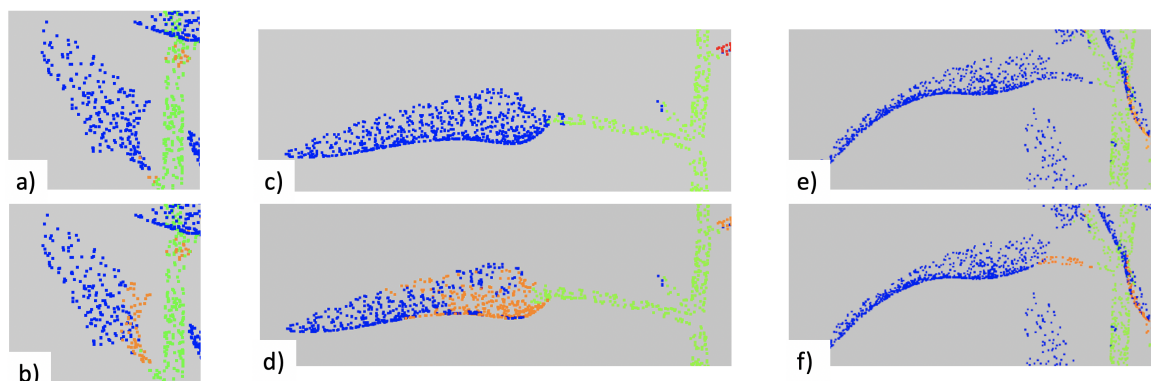


Figure 6.5: Défauts de la segmentation (a, c, e) et impact de l'étape d'amélioration (b, d, f). Les limbes sont en bleu, les tiges en vert et les pétioles en orange. Les figures a) et b) présentent la difficulté d'obtenir des frontières nettes lorsque le pétiole est très petit. Les figures e) et f) présentent une erreur de sous-segmentation d'un limbe et d'un pétiole, erreur corrigée grâce à l'étape d'amélioration.

Pour les plantes synthétiques, la procédure d'amélioration n'améliore pas suffisamment les résultats par rapport aux nouvelles erreurs qu'elle fait apparaître. Néanmoins, nous allons voir que l'impact de cette étape est plus positif sur les plantes réelles.

Sur les chénopodes réels

Nous avons testé notre chaîne de traitement sur les 5 chénopodes réels dont la segmentation à la main a été effectuée à la main. Nous avons donc pu réaliser une segmentation de référence qui contenait toutes les classes que nous obtenons grâce à notre chaîne de traitement. Les résultats quantitatifs sont disponibles en table 6.2.

Classe	Sans amélioration			Avec amélioration		
	Re	Pr	IoU	Re	Pr	IoU
Limbe	85	76	70	90	79	73
Tige principale	90	78	73	94	76	73
Apex	72	90	69	73	95	71
Pétiole	52	38	26	54	39	29
Branche	52	18	16	58	18	16

Table 6.2: Résultats moyens (en %) de la segmentation sémantique sur le jeu de données des 5 chénopodes réels. Les résultats sont présentés sans et avec l'étape d'amélioration expliquée en section 5.2.5.

Globalement, les résultats obtenus sur les chénopodes réels sont comparables aux résultats du jeu de données synthétique si l'on prend en compte que l'on évalue

sur des classes plus petites. Le macro F_1 -score est de 78% sans étape d'amélioration et de 80% avec l'étape. Les résultats quantitatifs sont légèrement diminués pour les limbes et les tiges et branches mais on peut constater une amélioration grâce à l'étape de correction. Contrairement aux résultats sur les plantes virtuelles, les scores de rappel sont souvent plus élevés que les scores de précision. L'algorithme a donc identifié dans ces classes un certain nombre de points qui n'étaient pas dans ces dernières. On pourrait éventuellement expliquer cette particularité par la plus grande variabilité des formes d'organes que proposent les plantes réelles. La nouvelle classe apex, au contraire, dispose d'un score de rappel élevé par rapport à son score de précision. Certains point des apex ont été manqués et classés dans un autre organe. Cette situation est illustrée dans la Figure 6.6 où l'on constate que l'ensemble de l'apex expertisé en g) a été découpé dans notre algorithme dont le résultat est visible en h) en un pétiole, des limbes, et un ensemble plus restreint a été sélectionné en apex. Les pétioles et les branches ont des scores similaires à ceux constatés pour les plantes virtuelles. Néanmoins, pour les chénopodes réels, on notera que les branches contiennent moins de points que les pétioles. Les problèmes de définitions décrits plus tôt pour les pétioles s'appliquent donc ici pour les branches et sont visibles en Figure 6.6 où une petite partie des pétioles a été détectée en j) mais pas suffisamment de points ont été classifiés pour correspondre à la segmentation de référence en i).

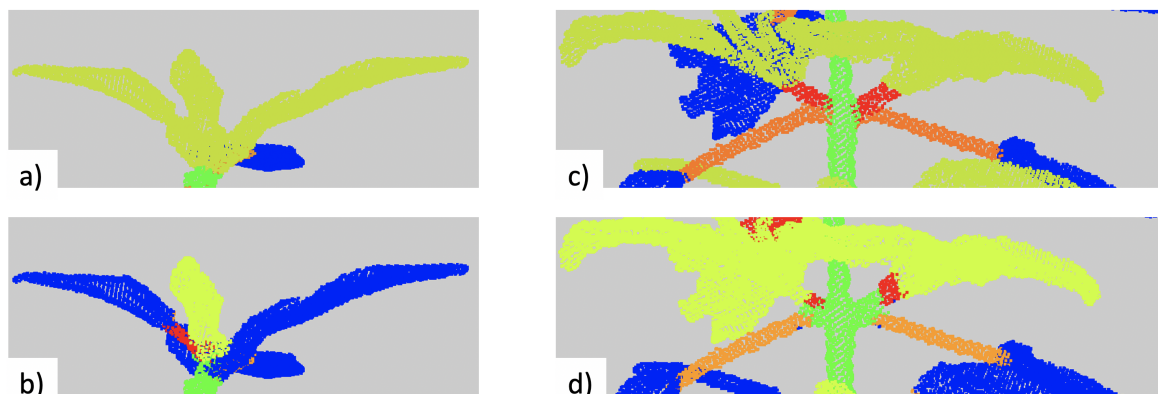


Figure 6.6: Segmentations de références (a, c) et segmentations obtenues grâce à la chaîne de traitement proposée (b, d). Les limbes sont en bleu, les tiges en vert et les pétioles en orange. Dans le premier exemple, la segmentation de référence (a) est moins segmentée que la segmentation obtenue (b). La segmentation issue de la chaîne de traitement présente un niveau de détail plus important. Dans le second exemple, les deux branches ont été détectées dans le résultat (d) mais n'englobent pas suffisamment de points du nuage par rapport à la référence (c).

Sur les tomates réelles (Phéno4D)

Afin de tester les capacités de cette chaîne de traitement à segmenter d'autres espèces de plantes, nous avons évalué les résultats sur 14 nuages de points de tomate

issus du jeu de données Phéno4D (SCHUNCK et al., 2021). Nous n'avons pas pris dans ces 14 nuages des plantes au stade de plantules et nous sommes donc limités à des plantes ayant déjà développé des branches. Seules deux classes étaient exploitables pour nous dans ce jeu de données, une classe limbe et une classe tige, autrement dit, une classe avec toutes les parties linéaires de la plantes et une classe avec les parties plus planes. Comme réalisé précédemment, nous avons fusionné les classes dont nous disposons en sortie de notre algorithme pour nous conformer à la segmentation de référence. Nous n'avons pas adapté les étapes sémantiques à la tomate étant donné que nous devons de toute façon fusionner les classes en sortie et nous ne présentons donc pas ici les résultats de l'étape d'amélioration.

Les résultats de notre comparaison sont disponibles en Table 6.3. Malgré une différence notable entre l'architecture d'un plant de tomate et celle d'un chénopode, la chaîne de traitement nous permet d'obtenir un F_1 -score de 85%. On note qu'à l'instar des résultats précédents, le score sur les limbes est élevé. Le score sur les tiges est plus faible. Étant donné que nous n'avons que deux classes, nous savons que la quantité de points mal classés dans une classe correspond directement à des points manqués dans l'autre classe. Ainsi, lorsque l'on constate un rappel de 59% pour les tiges, on peut en déduire que les points manquants étaient dans la classe tige. Ces points sont visibles dans les 15 % d'erreur de précision constatés dans la classe limbe. Cela permet également de souligner que les scores utilisés sont tributaires du nombre de points dans la classe en question. Ces résultats peuvent être mis côte à côte avec ceux obtenus par SCHUNCK et al., 2021, dans lequel les auteurs ont utilisé trois réseaux neuronaux différents pour segmenter le jeu de données complet disponible. Les IoU obtenus pour les classes limbe et tige étaient respectivement de 84 % et 12 pour PointNet, 89 % et 20 % pour PointNet++ et 99 % et 87 % pour LatticeNet. Nous avons utilisé un jeu de données plus petit que celui utilisé dans l'article de SCHUNCK et al., 2021, néanmoins, sur ce jeu de données réduit, nous pouvons constater que notre algorithme a des résultats supérieurs pour la détection de la tige par rapport à PointNet et PointNet++. Notre algorithme ne semble pas aussi performant que LatticeNet. Cette tendance serait à confirmer en comparant les résultats sur deux jeux de données parfaitement identiques.

Nous avons pu nous assurer que la qualité de la segmentation ne dépendait pas du stade de développement de la plante en Figure 6.7. On peut constater que sur les deux plants de tomates étudiés, disposant chacun de 7 étapes de croissances, on ne peut pas remarquer de tendance dans la qualité du score de segmentation.

Classe	Notre algorithme (sur un jeu de données Pheno4D réduit)			PointNet	PointNet++	LatticeNet
	Re	Pr	IoU	IoU	IoU	IoU
Feuille	90	85	83	84	89	99
Tige	59	75	51	12	20	87

Table 6.3: Résultats moyens (en %) de la segmentation sémantique en deux classes effectuée sur 14 plants de tomates. Résultats de la segmentation sémantique obtenue par SCHUNCK et al., 2021 avec trois réseaux neuronaux différents et sur le jeu de données de Pheno4D entier.

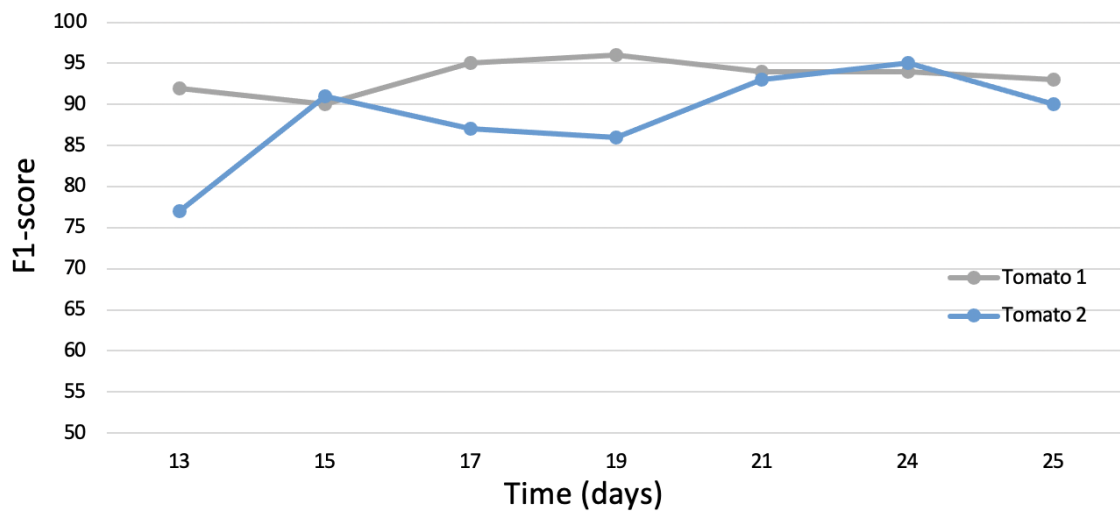


Figure 6.7: F_1 -score sur la segmentation de deux plants de tomates différents à 7 stades de croissance différents.

6.3.3 Évaluation de la segmentation par instance

Afin d'évaluer la segmentation par instance, nous avons utilisé le jeu de données des chénopodes réels. Nous avons dans un premier temps utilisé les métriques définies en section 6.3.1 et dont les valeurs obtenues pour notre jeu de données sont présentées en Table 6.4. On peut noter immédiatement que la segmentation par instance est un peu améliorée, quel que soit le score d'évaluation utilisé, lorsque l'on applique l'étape d'amélioration de la segmentation. Cela concorde avec les précédents résultats sur la segmentation sémantique des chénopodes réels. On constate donc que la segmentation par instance est également raffinée. Dans notre cas, la taille des instances peut être très variable en terme de quantité de points (d'une branche qui démarre à peine sa croissance à un limbe très développé), le score le plus pertinent est donc le score "AMI". On peut constater qu'il se situe entre 0,89 et 0,91. Il s'agit d'un score très correct. Cela indique que la majorité des points qui doivent être regroupés dans une même

partie, le sont. On obtient un score d'homogénéité très élevé (0,95). Cela signifie que les parties réalisées par notre algorithme tendent à contenir un ensemble cohérent de points devant bien être regroupés d'après la segmentation par instance de référence. Le score d'exhaustivité (completeness) est très élevé également mais légèrement plus faible. Autrement dit, les parties sont très cohérentes mais n'incluent pas toujours tous les points nécessaires. Elles ne sont pas complètes.

Score	Sans amélioration	Avec amélioration
Rand	0.96	0.97
Rand ajusté	0.79	0.82
AMI	0.89	0.91
Exhaustivité	0.85	0.87
Homogénéité	0.94	0.95

Table 6.4: Moyenne des scores obtenus pour chaque segmentation par instance calculée pour les 5 chénopodes virtuels traités. Les scores ont été calculés avec et sans l'étape d'amélioration de la segmentation expliquée en section 5.2.5.

Enfin, nous avons voulu évaluer la qualité de définition des instances. Pour ce faire, nous avons sélectionné les 28 limbes des 5 chénopodes réels obtenus avec l'étape d'amélioration. Nous les avons associés aux limbes de la segmentation par instance de référence et avons pu ainsi utiliser les mêmes métriques d'évaluation que celles utilisées pour la segmentation sémantique. Nous avons obtenu un score moyen sur toutes les limbes de rappel de 97%, de précision de 99% et un *mIoU* de 97%. Cela montre que l'algorithme est capable de correctement délimiter les limbes.

6.3.4 Extraction de traits phénotypiques pour l'exploitation agricole

A partir de la segmentation fournie par notre chaîne de traitement, il est possible de mesurer certains traits phénotypiques classiques. Nous avons sélectionné la hauteur de tige, l'aire foliaire totale de la plante et l'indice de surface foliaire (LAI : leaf area index). Ils nécessitent de disposer d'une segmentation par instance, d'une identification des limbes ainsi que d'une segmentation de la tige principale. Ces traits sont utiles pour déterminer le stade de la croissance de la plante. La hauteur de la tige permet de faire état du niveau de croissance primaire de la plante. L'indice de surface foliaire est généralement défini pour des récoltes et correspond à la surface foliaire totale au dessus d'une unité de surface de sol. C'est un indice sans dimension allant de 0 pour un sol nu, à plusieurs unités pour des canopées très denses. Cet indice est particulièrement pertinent lorsque l'on s'intéresse à l'état physiologique de la plante.

Plante	Surface boîte englobante (mm ²)	Surface totale des limbes (mm ²)		LAI		Hauteur tige principale (mm)	
		Vérité terrain	Segmenté	Vérité terrain	Segmenté	Vérité terrain	Segmenté
Chenopodium 1	6397,1	1216,9	1172,6	0,19	0,18	70,3	71,0
Chenopodium 2	5117,6	1420,2	1395,4	0,28	0,27	71,8	74,9
Chenopodium 3	4915,8	1745,4	1734,9	0,36	0,35	41,4	44,1
Chenopodium 4	8655,9	4060,7	4008,1	0,47	0,46	62,7	63,5
Chenopodium 5	6254,1	1895,7	1887,3	0,30	0,30	70,3	69,0

Table 6.5: Comparaison des surfaces de limbes totales et des indices de surface foliaire entre la segmentation en sortie de la chaîne de traitement et la segmentation de référence. Comparaison de la hauteur de tige principale obtenue à partir de la segmentation de référence et de la segmentation résultat de la chaîne de traitement.

En effet, cet indice conjugue la hauteur de la plante et la taille des feuilles à différents niveaux. Il est indispensable pour étudier la capacité de photosynthèse de la plante.

Nous avons voulu évaluer si la qualité de la segmentation était suffisante pour extraire ces traits phénotypiques sur les 5 chénopodes réels pour lesquels une segmentation de référence était disponible. La hauteur de la tige a été estimée en prenant le point le plus haut et le point le plus bas de la tige principale telle que définie par la segmentation (Table 6.5). L'écart moyen entre la hauteur de tige calculée sur la segmentation de référence et la segmentation résultat est inférieure à 2,3%. Cette méthode semble donc suffisante pour cette mesure.

Afin d'estimer l'indice de surface foliaire, nous avons d'abord généré une surface de Bézier (BOUKHANA et al., 2022) sur chaque instance de limbe, de la segmentation de référence et de la segmentation résultat. Les 28 limbes mesurées sur les 5 plantes avaient une surface moyenne de 377 mm² et nous avons constaté un écart-type de 7,85 mm² entre les surfaces calculées à partir de la segmentation de référence et celles obtenues à partir de la segmentation résultat. Nous avons additionné les surfaces de tous les limbes de chaque plante pour obtenir sa surface foliaire totale (Table 6.5). Enfin, nous avons divisé la surface foliaire totale de chaque plante par la surface au sol de sa boîte englobante. Cette surface nous donne une approximation de la surface au sol qu'occuperait la plante dans un champ.

Ces quelques résultats montrent que des traits phénotypiques peuvent être extraits grâce à la segmentation en sortie de notre chaîne de traitement. L'écart moyen relatif est de 2,3 % pour la hauteur de tige et de -1,5 % pour la surface foliaire. L'écart-type relatif est de 3,7% pour la hauteur de plante et de 2,3% pour les surfaces foliaires.

6.4 Discussion et conclusion

La chaîne de traitement proposée dans ce chapitre répond au besoin d'obtenir une segmentation par instance et sémantique d'un nuage des points 3D de plantes. Nous avons d'abord évalué la segmentation sémantique avec des outils classiques de la littérature, les scores de précision, de rappel, ainsi entre autre que le $F1$ -score. On note que nous avons dû adapter notre segmentation sémantique à la classification dont nous disposons dans les jeux de données de chénopodes virtuels et de plants de tomates. Seule la segmentation de référence des chénopodes réels correspondait à nos définitions de classes en sortie de la chaîne de traitement. Quelque soit le jeu de données, on peut noter que les limbes sont les classes considérées comme les mieux segmentées par les scores. Les pétioles et les branches sont les organes les plus petits et les plus difficiles à segmenter. Bien qu'utilisés dans la plupart des articles (BOOGAARD et al., 2022, SCHUNCK et al., 2021, TURGUT et al., 2022), on peut poser ici la question de la pertinence de ces scores lorsque les classes ont des tailles de points très différentes. L'impact de la boucle de correction est notamment difficilement visible et évaluable avec les scores que nous avons mis en place. Bien que cette boucle permette de récupérer des organes qui étaient fusionnés ensemble, lorsque l'on recalcule les scores avec les points mieux classifiés, l'amélioration est divisée puisque répartie dans chaque classe. L'amélioration serait plus visible à une autre échelle, telle que celle de l'arbre sémantique.

Les jeux de données des tomates réelles et des chénopodes virtuels ne proposent que des segmentations de référence sémantiques. L'évaluation de la segmentation par instance a donc été réalisée grâce au jeu de données des chénopodes réels. D'une part, nous avons utilisé des indices statistiques pour évaluer la qualité du regroupement de points effectuée. D'autre part, nous avons considéré l'instance des limbes et évalué la qualité de définitions de ces instances. Les premiers comme les derniers ont été très encourageant mais gagneraient à être confirmés sur un jeu de données plus conséquent. De la même manière, le calcul de traits phénotypiques sur le jeu de données des chénopodes réels a présenté des résultats corrects (avec des écarts types relatifs sur ces valeurs inférieurs à 4%). Il serait bon de pouvoir confirmer ces résultats sur un jeu de données plus grand. L'étiquetage de jeux de données pour effectuer des évaluations représente un réel enjeu qui se heurte à deux problématiques très concrètes : l'opération est chronophage et demande de la ressource humaine.

Conclusion

Les segmentations par instance et sémantique automatiques de nuages de points 3D de plantes représentent une problématique importante en recherche. En effet, nous avons pu voir au cours du chapitre 2 que pour déterminer le phénotype architectural d'une plante, il était nécessaire d'effectuer des mesures sur un grand nombre de plantes. La mesure de traits phénotypiques à grand rendement nécessite d'obtenir un modèle 3D de ces plantes. Aujourd'hui, la plupart des systèmes d'acquisitions abordables et précis fournissent en sortie un nuage de points 3D. Lorsque les traits à mesurer sont dits complexes (PAULUS, 2019), ils nécessitent que les organes de la plante aient été segmentés et identifiés dans le nuage. Afin de comprendre cette opération et de rendre compte des solutions existantes à ce problème dans la littérature, nous avons dans le chapitre 3 effectué un tour d'horizon des possibilités. Nous avons pu constater que lorsqu'il s'agit de segmentation par instance, une grande problématique émerge, peu importe la donnée et l'objet à l'étude. La détermination d'attributs discriminants est essentielle pour pouvoir segmenter un jeu de données. Bien que certaines méthodes spécifiques aient émergé dans le domaine des nuages de points de plantes, ces dernières sont souvent limitées et sont composées d'un agglomérat de différentes méthodes pour gérer la variabilité des espèces de plantes et les caractéristiques des différents organes de la plante. Pour la segmentation sémantique, nous avons pu constater que l'utilisation de connaissances associées à l'objet est indispensable pour déterminer les classes et peut potentiellement permettre de détecter certaines erreurs de la segmentation sémantique.

Afin d'obtenir un attribut discriminant pertinent, nous nous sommes intéressés dans le chapitre 4, à la construction d'un graphe de similarité à partir du nuage de points et au spectre de ce graphe. Suite à nos observations sur le vecteur de Fiedler, nous avons calculé un champ de vecteurs, le gradient directionnel moyen du vecteur de Fiedler. Nous avons ainsi obtenu deux attributs discriminants : la norme (permettant de différencier les limbes) et les directions (permettant de séparer les organes linéaires entre eux) de ce champ. Nous avons ensuite présenté des algorithmes de segmentation non supervisée permettant d'exploiter ces attributs. Rapidement limités par le graphe de similarité représentant la topologie à l'échelle des points et pour obtenir une segmentation sémantique, nous avons dans le chapitre 5 introduit un nouvel outil, le graphe quotient. Ce nouveau graphe permet une vision à l'échelle de l'instance et non plus du point. En nous appuyant sur cette structure, nous classifions les instances de

points organe par organe. Nous commençons par différencier les limbes des apex, puis nous détectons la tige principale, et enfin les pétioles et les branches. Cette dernière étape nous permet également de simplifier le graphe quotient et de l'affiner de façon à ce que chacun de ses nœuds représente un organe de la plante et que chaque arête du graphe quotient soit une liaison botanique plausible entre les différents organes. Il s'agit alors d'un arbre sémantique et cette structure nous permet de détecter les incohérences botaniques et de potentiellement les corriger.

Après avoir présenté un certain nombre d'outils dans les chapitres 4 et 5, nous présentons dans le chapitre 6 une chaîne de traitement allant du nuage de points initial jusqu'au nuage de points segmenté par instance et classifié, ainsi qu'à la réalisation de l'arbre sémantique décrivant la structure de la plante. Nous avons évalué cette chaîne de traitement avec trois jeux de données différents composés de chénopodes réels et virtuels ainsi que de plants de tomates. De part la diversité entre les segmentations disponibles pour ces jeux de données, la segmentation par instance n'a été évaluée que sur les chénopodes réels et la segmentation sémantique sur les trois jeux. Enfin, nous avons calculé des traits phénotypiques à partir de notre segmentation et de la segmentation de référence sur les nuages de points de chénopodes réels. Nous avons pu constater que la chaîne de traitement obtenait de bon résultats avec des $F1$ -scores globaux de 80% sur la segmentation sémantique et sur la segmentation par instance. On parvient en particulier à bien détecter les feuilles et leurs frontières. Enfin les segmentations obtenues sur les chénopodes réels semblent suffisamment précises pour permettre d'estimer avec une bonne précision des traits phénotypiques à valeur agronomique (LAI, surfaces foliaires, hauteur de tige).

Bien que beaucoup de méthodes sont aujourd'hui basées sur l'apprentissage par réseaux neuronaux, nous pensons qu'il est intéressant de continuer à s'intéresser à des méthodes algorithmiques telles que celle présentée dans cette thèse. En effet, bien que les réseaux neuronaux présentent des résultats très bons sur la reconnaissance des organes de la plante (par exemple dans BOOGAARD et al., 2022), ces approches présentent des limitations. Dans un premier temps, le nuage de points doit être drastiquement ré-échantillonné, au risque de perdre des détails importants comme les pétioles ou les fleurs. Ensuite, il est nécessaire de disposer d'une quantité très importante de données déjà segmentées et cette étape est chronophage et demande des ressources humaines importantes. Enfin, aucune garantie n'est donnée sur la cohérence botanique de la segmentation obtenue. Des limbes peuvent être accolées à la tige principale par exemple. On notera également qu'un réseau neuronal entraîné sur une plante en particulier ne sera pas forcément adapté à une autre espèce de plantes. La chaîne de traitement que nous proposons permet de ne pas passer par une

étape d'apprentissage. Les attributs basés sur le spectre du graphe semblent adaptables à différentes variétés de plantes si l'on en croit les premiers résultats sur les plants de tomates obtenus ainsi que sur les chénopodes. L'approche est relativement rapide et nous avons pu constater sur le jeu de données des tomates que les résultats sont aussi bons que ceux de réseaux neuronaux non spécialisés dans les plantes comme PointNet et PointNet ++. Cette approche paraît donc plus abordable que les réseaux neuronaux dans les situations où on ne dispose pas du temps nécessaire à la création d'un jeu de données d'apprentissage conséquent et à l'entraînement du réseau neuronal, ou encore de la puissance de calcul nécessaire.

Nous pensons qu'il est encore possible d'améliorer cette chaîne de traitement de plusieurs façons. Concernant les attributs calculés, la norme et les directions du GDM du vecteur de Fiedler, nous nous sommes arrêtés dans cette thèse à l'utilisation d'algorithmes de segmentation non supervisés pour les exploiter. En effet, nous avons utilisé l'algorithme des K-means ainsi qu'un algorithme de division/fusion à l'échelle du graphe quotient. Il serait pourtant intéressant d'observer les valeurs et le comportement de ces attributs sur des plantes déjà classées et de tester des algorithmes d'apprentissage supervisé sur ces données comme dans la chaîne de traitement proposée par WEINMANN et al., 2015. Cela permettrait notamment de prendre en compte la structure du graphe de similarité et donc d'éviter les étapes de croissance par région et de filtrage que nous avons dû déployer. Potentiellement, cela permettrait de limiter des erreurs de sous-segmentation que nous avons pu observer après l'utilisation des K-means. L'observation du GDM du vecteur de Fiedler sur un grand nombre de variétés de plantes paraît également indispensable pour pouvoir potentiellement justifier de la pertinence de cet attribut pour les plantes de façon plus générale que ce qui a été fait au cours de cette étude.

Dans notre méthode, nous aboutissons également à la construction d'un objet, l'arbre sémantique. Ce dernier est un arbre au sens informatique du terme. Chacun de ses nœuds représente un organe de la plante et chaque arête de ce graphe un lien biologiquement plausible entre ces deux organes. Nous utilisons cet arbre dans cette thèse pour contrôler la segmentation. Néanmoins, nous pensons qu'il serait utile de pouvoir mettre en place une méthode d'évaluation de cet arbre sémantique. Nous avons évalué la segmentation par instance et la segmentation sémantique obtenue sur différents nuages de points mais nous n'avons pas évalué la qualité de l'arbre sémantique. L'évaluation d'un tel arbre sémantique demanderait d'une part de disposer d'un arbre sémantique de référence pour chaque nuage de points de plantes sur lesquels on veut évaluer l'objet mais également de disposer d'une métrique permettant de comparer les deux arbres. Il est envisageable de s'inspirer des distances

d'édition entre deux graphes comme proposé dans ABU-AISHEH et al., 2015 ou encore directement entre deux arbres (Ferraro et Godin, 2000, BILLE, 2005). Ces distances d'éditions permettraient potentiellement de prendre en compte directement le niveau de classification de la segmentation de référence. En effet, comme on a pu le constater dans le chapitre 6, tous les jeux de données disponibles ne classifiaient pas les points dans autant de classes ou avec les mêmes définitions d'organes (exemple des apex) que ce que notre chaîne de traitement fournissait. Par conséquent nous avons été fréquemment amené à changer les classes de nos nuages pour nous conformer aux classe de la segmentation d'origine. Nous pensons que cette étape pourrait être plus facilement gérée s'il suffisait de définir dans une distance d'édition que le changement d'un nœud de l'arbre d'apex en limbe ne représente pas un coût.

Une fois la structure de l'arbre sémantique évaluée et validée, cet arbre sémantique pourrait constituer un outil de description de la plante à l'étude. Il donne par nature l'organisation de la plante, le nombre de limbes, de pétioles. Au même titre que nous avons attaché des attributs aux nœuds du graphe quotient, il pourrait en être de même pour l'arbre sémantique. La taille de chaque organe, par exemple, pourrait être disponible comme un attribut de nœud de l'arbre. Autrement dit, nous pensons que cet arbre sémantique pourrait être utilisé directement pour comparer la structure de différents phénotypes de plantes entre elles, éventuellement détecter des anomalies, d'autant plus qu'il existe déjà un nombre important d'outils de manipulation de graphe (par exemple dans AZAIS et al., 2019). Cela ouvre également à la possibilité de travailler à différentes échelles de la plante pour observer ses traits phénotypiques. Nous savons par exemple qu'une feuille est constituée d'un limbe et d'un pétiole. Une fois l'arbre sémantique correctement réalisé, il devient également possible pour l'utilisateur de préciser s'il souhaite faire la différence entre limbe ou pétiole ou ne considérer qu'un seul nœud étiqueté en feuille.

Enfin, pour que notre chaîne de traitement soit exploitable directement par des botanistes ayant acquis des nuages de points 3D de plantes et qui souhaitent mesurer des traits spécifiques, il serait intéressant de pouvoir trouver une solution pour exprimer simplement sous la forme d'un graphe la façon dont on attend que les organes doivent se positionner entre eux. La méthode telle que décrite actuellement est adaptée à la structure du chénopode, elle doit pouvoir être adaptable à d'autres espèces. Étant donné que toute la classification dans notre chaîne de traitement s'effectue sur une base de plus courts chemins et de graphes, il serait intéressant de pouvoir travailler à la mise en place automatique des bonnes pondérations et de l'ordre dans lequel déterminer les organes en fonction de la plante à l'étude et de ses caractéristiques biologiques. La partie algorithmique basée sur le graphe quotient pourrait également

être utilisées en sortie d'autres algorithmes de segmentation par instance car elle n'est pas dépendante de la méthode utilisée en amont. Elle pourrait présenter de ce fait, une façon de contrôler la cohérence botanique d'autres algorithmes et pourquoi pas, de réseaux neuronaux.

Publications et conférences :

Mirande, K., Hétroy-Wheeler, F., et Godin, C. (2020). High-precision 3D segmentation of plants combining spectral clustering and quotient graph techniques : a multi-level approach. In *9th International Conference on Functional-Structural Plant Models*.

Mirande, K., Godin, C., Tisserand, M., Charlaix, J., Besnard, F., et Hétroy-Wheeler, F. (2022). A Graph-Based Approach for Simultaneous Semantic and Instance Segmentation of Plant 3D Point Clouds. *Frontiers in Plant Science*. (accepté, en cours d'édition)

Bibliographie

- ABU-AISHEH, Zeina, Romain RAVEAUX, Jean-Yves RAMEL et Patrick MARTINEAU (jan. 2015). « An Exact Graph Edit Distance Algorithm for Solving Pattern Recognition Problems ». In : *4th International Conference on Pattern Recognition Applications and Methods 2015*. Lisbon, Portugal (cité en page 108).
- AGISOFT, LLC (2014). « Agisoft Photoscan Pro ». In : *St. Petersburg, Russia. Available online at : <http://www.agisoft.com>* (cité en page 58).
- ANDREOTTI, Eleonora, Daniel REMONDINI, Graziano SERVIZI et Armando BAZZANI (2018). « On the multiplicity of Laplacian eigenvalues and Fiedler partitions ». In : *Linear Algebra and its Applications* 544, p. 206-222 (cité en page 49).
- ANKERST, Mihael, Markus M. BREUNIG, Hans-Peter KRIEGEL et Jörg SANDER (1999). « OPTICS : ordering points to identify the clustering structure ». en. In : *Proceedings of the 1999 ACM SIGMOD international conference on Management of data - SIGMOD '99*. Philadelphia, Pennsylvania, United States : ACM Press, p. 49-60 (cité en page 56).
- AUGUSTIN, Marco, Yll HAXHIMUSA, Wolfgang BUSCH et Walter G. KROPATSCH (juil. 2016). « A framework for the extraction of quantitative traits from 2D images of mature Arabidopsis thaliana ». en. In : *Machine Vision and Applications* 27.5, p. 647-661 (cité en page 13).
- AZAIS, Romain, Guillaume CERUTTI, Didier GEMMERLE et Florian INGELS (2019). « Treex : a Python package for manipulating rooted trees ». In : *Journal of Open Source Software* 4.38, p. 1351 (cité en page 108).
- BARLOW, PW (1989). « Meristems, metamers and modules and the development of shoot and root systems ». In : *Botanical Journal of the Linnean Society* 100.3, p. 255-279 (cité en pages 9, 10).
- BARTHÉLÉMY, Daniel et Yves CARAGLIO (mar. 2007). « Plant Architecture : A Dynamic, Multilevel and Comprehensive Approach to Plant Form, Structure and Ontogeny ». en. In : *Annals of Botany* 99.3, p. 375-407 (cité en pages 6, 8, 9).
- BARTHÉLÉMY, Daniel, Yves CARAGLIO et Evelyne COSTES (1997). « Architecture, gradients morphogénétiques et âge physiologique chez les végétaux ». In : *Modélisation et simulation de l'architecture des végétaux*, p. 89-136 (cité en page 8).

- BELL, Adrian D. (1991). *An Illustrated Guide to Flowering Plant Morphology*. en. 200 Madison Ave., New-York, NY 10016 : Oxford University Press (cité en pages 8, 9).
- BERGER, Matthew, Andrea TAGLIASACCHI, Lee M. SEVERSKY et al. (jan. 2017). « A Survey of Surface Reconstruction from Point Clouds : A Survey of Surface Reconstruction from Point Clouds ». en. In : *Computer Graphics Forum* 36.1, p. 301-329 (cité en pages 24, 25, 58, 74).
- BERTRAND, Sarah (déc. 2018). « Analyse d'images pour l'identification multi-organes d'espèces végétales ». Theses. Université de Lyon (cité en page 1).
- BESNARD, Fabrice (oct. 2011). « Study of the role of AHP6 in the control of phyllotaxis in *Arabidopsis thaliana* : robustness and spatio-temporal coordination in the development of self-organized organisms ». Theses. Ecole normale supérieure de lyon - ENS LYON, 346 p. (Cité en page 16).
- BILLE, Philip (2005). « A survey on tree edit distance and related problems ». In : *Theoretical computer science* 337.1-3, p. 217-239 (cité en page 108).
- BLAZÈRE, Mélanie (avr. 2016). « A variant of the spectral clustering method using l1 -penalty to promote sparse eigenvectors basis » (cité en page 48).
- BLUM, Harry et Roger N NAGEL (1978). « Shape description using weighted symmetric axis features ». In : *Pattern recognition* 10.3, p. 167-180 (cité en page 30).
- BOOGAARD, Frans P., Eldert J. van HENTEN et Gert KOOTSTRA (mar. 2022). « Improved Point-Cloud Segmentation for Plant Phenotyping Through Class-Dependent Sampling of Training Data to Battle Class Imbalance ». en. In : *Frontiers in Plant Science* 13, p. 838190 (cité en pages 32, 33, 104, 106).
- BOUDON, Frederic, Christophe PRADAL, Thomas COKELAER, Przemyslaw PRUSINKIEWICZ et Christophe GODIN (2012). « L-Py : An L-System Simulation Framework for Modeling Plant Architecture Development Based on a Dynamic Language ». In : *Frontiers in Plant Science* 3 (cité en pages 16, 92).
- BOUKHANA, Mélinda, Joris RAVAGLIA, Franck HÉTROY-WHEELER et Benoît De SOLAN (2022). « Geometric models for plant leaf area estimation from 3D point clouds : A comparative study ». In : *Graphics and Visual Computing*, p. 200057 (cité en page 103).
- BRANCH, WD et RO HAMMONS (1983). « Inheritance of a Micro Phenotype in Peanut 1 ». In : *Crop science* 23.6, p. 1045-1046 (cité en page 11).
- CARNÉ-CARNAVALET, C. de (2020). *Le maraîchage sur petite surface : La French Method : une agriculture urbaine ou périurbaine*. Editions de Terran (cité en page 15).
- CASELLA, Eric et Hervé SINOQUET (2003). « A method for describing the canopy architecture of coppice poplar with allometric relationships ». In : *Tree Physiology* 23.17, p. 1153-1170 (cité en page 58).
- CHAUDHURY, Ayan, Frederic BOUDON et Christophe GODIN (2020). « 3D plant phenotyping : All you need is labelled point cloud data ». In : *European Conference on Computer Vision*. Springer, p. 244-260 (cité en pages 16, 92).

- CHAUDHURY, Ayan et Christophe GODIN (2020a). « Geometry Reconstruction of Plants ». In : *Intelligent Image Analysis for Plant Phenotyping*. CRC Press, p. 119-142 (cité en page 30).
- (juin 2020b). « Skeletonization of Plant Point Cloud Data Using Stochastic Optimization Framework ». en. In : *Frontiers in Plant Science* 11 (cité en pages 2, 30).
- CHAUDHURY, Ayan, Christopher WARD, Ali TALASAZ et al. (2018). « Machine vision system for 3D plant phenotyping ». In : *IEEE/ACM transactions on computational biology and bioinformatics* 16.6, p. 2009-2022 (cité en page 13).
- CHEBROLU, Nived, Federico MAGISTRI, Thomas LÄBE et Cyrill STACHNISS (fév. 2021). « Registration of spatio-temporal point clouds of plants for phenotyping ». In : *PLOS ONE* 16.2. Publisher : Public Library of Science, p. 1-25 (cité en page 31).
- CHÉNÉ, Yann, David ROUSSEAU, Philippe LUCIDARME et al. (2012). « On the use of depth camera for 3D phenotyping of entire plants ». In : *Computers and Electronics in Agriculture* 82, p. 122-127 (cité en page 1).
- COMMUNITY, Blender Online (2018). *Blender - a 3D modelling and rendering package*. Blender Foundation. Stichting Blender Foundation, Amsterdam (cité en page 52).
- ELLIS, Beth, Douglas DALY, Leo HICKEY et al. (juin 2009). *Manual of Leaf Architecture*. Cornell University Press CABI (cité en page 6).
- ELNASHEF, Bashar, Sagi FILIN et Ran Nisim LATI (jan. 2019). « Tensor-based classification and segmentation of three-dimensional point clouds for organ-level plant phenotyping and growth analysis ». en. In : *Computers and Electronics in Agriculture* 156, p. 51-61 (cité en page 32).
- ESTER, Martin, Hans-Peter KRIEGEL, Jörg SANDER, Xiaowei XU et al. (1996). « A density-based algorithm for discovering clusters in large spatial databases with noise. » In : *kdd*. T. 96. 34, p. 226-231 (cité en page 56).
- ESTIVILL-CASTRO, Vladimir (2002). « Why so many clustering algorithms : a position paper ». In : *ACM SIGKDD explorations newsletter* 4.1, p. 65-75 (cité en page 24).
- FERRARO, Pascal et Christophe GODIN (2000). « A distance measure between plant architectures ». In : *Annals of Forest Science* 57.5, p. 445-461 (cité en page 108).
- FIEDLER, Miroslav (1973). « Algebraic connectivity of graphs ». eng. In : *Czechoslovak Mathematical Journal* 23.2, p. 298-305 (cité en page 49).
- (1989). « Laplacian of graphs and algebraic connectivity ». EN. In : *Banach Center Publications* 25.1. Publisher : Institute of Mathematics Polish Academy of Sciences, p. 57-70 (cité en pages 41, 49).
- FORTIER, Jean-François (2022). *Forme foliaire : définition et explications*. fr. <https://www.aquaportail.com/definition-14812-forme-foliaire.html> (cité en page 7).
- FURBANK, Robert T. et Mark TESTER (déc. 2011). « Phenomics – technologies to relieve the phenotyping bottleneck ». en. In : *Trends in Plant Science* 16.12, p. 635-644 (cité en page 10).

- GELARD, William, Michel DEVY, Ariane HERBULOT et Philippe BURGER (juin 2017). « Segmentation de nuages de points 3D pour le phénotypage de tournesols ». In : *16ème Journées Francophones des Jeunes Chercheurs en Vision par Ordinateur (ORASIS 2017)*. Colleville-sur-Mer, France : GREYC, 8p. (Cité en pages 31, 34).
- GHAHREMANI, Morteza, Kevin WILLIAMS, Fiona M. K. CORKE et al. (2021). « Deep Segmentation of Point Clouds of Wheat ». In : *Frontiers in Plant Science* 12, p. 429 (cité en page 31).
- GODIN, C. et Y. CARAGLIO (mar. 1998). « A Multiscale Model of Plant Topological Structures ». en. In : *Journal of Theoretical Biology* 191.1, p. 1-46 (cité en page 35).
- GODIN, Christophe (juin 2000). « Representing and encoding plant architecture : A review ». en. In : *Annals of Forest Science* 57.5, p. 413-438 (cité en page 1).
- GOLOVINSKIY, Aleksey et Thomas FUNKHOUSER (2009). « Min-cut based segmentation of point clouds ». In : *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. IEEE, p. 39-46 (cité en page 27).
- GPL SOFTWARE (2022). *CloudCompare*. Version 2.12.4 (cité en page 52).
- GRILLI, E., F. MENNA et F. REMONDINO (fév. 2017). « A Review of Point Clouds Segmentation and Classification Algorithms ». en. In : *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W3*, p. 339-344 (cité en pages 24, 25, 27, 29).
- GROFF, Paul A et Donald R KAPLAN (1988). « The relation of root systems to shoot systems in vascular plants ». In : *The Botanical Review* 54.4, p. 387-422 (cité en pages 5, 6).
- GUINARD, S. et L. LANDRIEU (mai 2017). « Weakly Supervised Segmentation-Aided Classification of Urban Scenes from 3D LIDAR Point Clouds ». en. In : *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-1/W1*, p. 151-157 (cité en page 27).
- GUO, Yulan, Hanyun WANG, Qingyong HU et al. (2020). « Deep learning for 3d point clouds : A survey ». In : *IEEE transactions on pattern analysis and machine intelligence* 43.12, p. 4338-4364 (cité en pages 28, 32).
- HACKENBERG, Jan, Heinrich SPIECKER, Kim CALDERS, Mathias DISNEY et Pasi RAUMONEN (nov. 2015). « SimpleTree —An Efficient Open Source Tool to Build Tree Models from TLS Clouds ». en. In : *Forests* 6.12, p. 4245-4294 (cité en page 30).
- HEIN, Matthias (2006). « Uniform convergence of adaptive graph-based regularization ». In : *International Conference on Computational Learning Theory*. Springer, p. 50-64 (cité en page 43).
- HÉTROUY-WHEELER, Franck, Eric CASELLA et Dobrina BOLTCHIEVA (juil. 2016). « Segmentation of tree seedling point clouds into elementary units ». en. In : *International Journal of Remote Sensing* 37.13, p. 2881-2907 (cité en pages 2, 29, 32, 34, 35, 37, 58).

- HUAN, ZHAO, WANG JINGLU, LIAO SHENGJIN et al. (2021). « Study on the Micro-Phenotype of Different Types of Maize Kernels Based on Micro-CT ». In : *Smart Agriculture* 3.1, p. 16 (cité en page 11).
- INRA (2000). *HYPPA Accueil*. <https://www2.dijon.inrae.fr/>. Encyclopedie en ligne (cité en page 20).
- JAIN, Anil K, M Narasimha MURTY et Patrick J FLYNN (1999). « Data clustering : a review ». In : *ACM computing surveys (CSUR)* 31.3, p. 264-323 (cité en pages 23, 24).
- JANE, A. K. et R. C. DUBES (1998). « Algorithm of clustering data ». In : *PrenticeHall, Englewood Cliffs, NJ* (cité en pages 24, 37).
- KIM, HB et G SOHN (2011). « Random forests based multiple classifier system for power-line scene classification ». In : *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci* 38, p. 253-258 (cité en page 28).
- LANDRIEU, Loic et Martin SIMONOVSKY (2018). « Large-scale point cloud semantic segmentation with superpoint graphs ». In : *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, p. 4558-4567 (cité en pages 34, 35, 62).
- LAROUSSE, Éditions (2022). *Larousse.fr : encyclopédie et dictionnaires gratuits en ligne*. <https://www.larousse.fr/> (cité en pages 5, 6, 23, 25).
- LAURENTINI, Aldo (1994). « The visual hull concept for silhouette-based image understanding ». In : *IEEE Transactions on pattern analysis and machine intelligence* 16.2, p. 150-162 (cité en page 17).
- LEVY, B. (2006). « Laplace-Beltrami Eigenfunctions Towards an Algorithm That "Understands" Geometry ». en. In : *IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06)*. Matsushima, Japan : IEEE, p. 13-13 (cité en page 44).
- LI, Dawei, Yan CAO, Guoliang SHI et al. (2019). « An Overlapping-Free Leaf Segmentation Method for Plant Point Clouds ». en. In : *IEEE Access* 7, p. 129054-129070 (cité en page 59).
- LI, Dawei, Guoliang SHI, Weijian KONG, Sifan WANG et Yang CHEN (2020). « A Leaf Segmentation and Phenotypic Feature Extraction Framework for Multiview Stereo Plant Point Clouds ». In : *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13, p. 2321-2336 (cité en page 32).
- LI, Lei, Qin ZHANG et Danfeng HUANG (nov. 2014). « A Review of Imaging Techniques for Plant Phenotyping ». en. In : *Sensors* 14.11, p. 20078-20111 (cité en pages 10, 11).
- LI, Zhenbo, Ruohao GUO, Meng LI, Yaru CHEN et Guangyao LI (2020). « A review of computer vision technologies for plant phenotyping ». In : *Computers and Electronics in Agriculture* 176, p. 105672 (cité en page 13).
- LIU, Zichu, Qing ZHANG, Pei WANG, Zhen LI et Huiru WANG (2020). « Automated classification of stems and leaves of potted plants based on point cloud data ». In : *Biosystems engineering* 200, p. 215-230 (cité en pages 32, 34).

- LONCARIC, Sven (1998). « A survey of shape analysis techniques ». In : *Pattern recognition* 31.8, p. 983-1001 (cité en page 30).
- LUXBURG, Ulrike von (déc. 2007). « A tutorial on spectral clustering ». en. In : *Statistics and Computing* 17.4, p. 395-416 (cité en pages 41, 47, 48).
- MACHER, Hélène (2017). « Du nuage de points à la maquette numérique de bâtiment : reconstruction 3D semi-automatique de bâtiments existants ». Thèse de doct. Université de Strasbourg (cité en page 26).
- MACQUEEN, I (1967). « Some methods for classification and analysis of multivariate observations ». In : *Proceedings 5th Berkeley Symposium on Mathematical Statistics Problems*, p. 281-297 (cité en page 56).
- MEYEROWITZ, Elliot M (1987). « Arabidopsis thaliana ». In : *Annual review of genetics* 21.1, p. 93-111 (cité en page 19).
- MOHAR, Bojan (1997). « Some applications of Laplace eigenvalues of graphs ». In : *Graph symmetry*. Springer, p. 225-275 (cité en page 43).
- NARVAEZ, F. Yandun, G. REINA, M. TORRES-TORRITI, G. KANTOR et F. A. CHEEIN (déc. 2017). « A Survey of Ranging and Imaging Techniques for Precision Agriculture Phenotyping ». In : *IEEE/ASME Transactions on Mechatronics* 22.6, p. 2428-2439 (cité en page 13).
- NGUYEN, Anh et Bac LE (nov. 2013). « 3D point cloud segmentation : A survey ». en. In : *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. Manila, Philippines : IEEE, p. 225-230 (cité en pages 25, 26, 29, 37).
- NURUNNABI, Abdul, Geoff WEST et David BELTON (2015). « Outlier detection and robust normal-curvature estimation in mobile laser scanning 3D point cloud data ». In : *Pattern Recognition* 48.4, p. 1404-1419 (cité en page 27).
- OELLERMANN, Ortrud R et Allen J SCHWENK (1991). « The Laplacian spectrum of graphs ». In : *Graph Theory, c, Appl* 2, p. 871-898 (cité en page 40).
- PAPROKI, Anthony, Xavier SIRAUT, Scott BERRY, Robert FURBANK et Jurgen FRIPP (2012). « A novel mesh processing based technique for 3D plant analysis ». en. In : *BMC Plant Biology* 12.1, p. 63 (cité en page 34).
- PAULUS, Stefan (déc. 2019). « Measuring crops in 3D : using geometry for plant phenotyping ». en. In : *Plant Methods* 15.1 (cité en pages 12-14, 21, 105).
- PAULUS, Stefan, Jan DUPUIS, Anne-Katrin MAHLEIN et Heiner KUHLMANN (déc. 2013). « Surface feature based classification of plant organs from 3D laserscanned point clouds for plant phenotyping ». en. In : *BMC Bioinformatics* 14.1, p. 238 (cité en page 32).
- PEAUCELLE, Alexis, Halima MORIN, Jan TRAAS et Patrick LAUFS (2007). « Plants expressing a miR164-resistant CUC2 gene reveal the importance of post-meristematic maintenance of phyllotaxy in Arabidopsis ». In : *Development (Cambridge, England)* 134, p. 1045-1050 (cité en page 16).

- PEREZ-SANZ, Fernando, Pedro J NAVARRO et Marcos EGEEA-CORTINES (nov. 2017). « Plant phenomics : an overview of image acquisition technologies and image data analysis algorithms ». en. In : *GigaScience* 6.11 (cité en page 13).
- PIAO, Shilong, Qiang LIU, Anping CHEN et al. (2019). « Plant phenology and global climate change : Current progresses and challenges ». In : *Global change biology* 25.6, p. 1922-1940 (cité en page 1).
- POLANSKI, Andrzej et Marek KIMMEL (2007). *Bioinformatics*. en. Berlin ; New York : Springer (cité en page 56).
- POONIA, Amrita et Ashutosh UPADHAYAY (2015). « Chenopodium album Linn : review of nutritive value and biological properties ». In : *Journal of food science and technology* 52.7, p. 3977-3985 (cité en page 20).
- POTHEN, Alex, Horst D. SIMON et Kang-Pu LIOU (juil. 1990). « Partitioning Sparse Matrices with Eigenvectors of Graphs ». In : *SIAM Journal on Matrix Analysis and Applications* 11.3. Publisher : Society for Industrial and Applied Mathematics, p. 430-452 (cité en page 49).
- QI, Charles Ruizhongtai, Li Yi, Hao SU et Leonidas J GUIBAS (2017). « Pointnet++ : Deep hierarchical feature learning on point sets in a metric space ». In : *Advances in neural information processing systems* 30 (cité en pages 28, 33).
- RAGNI, Laura, Enric BELLES-BOIX, Markus GÜNL et Véronique PAUTOT (2008). « Interaction of KNAT6 and KNAT2 with BREVIPEDICELLUS and PENNYWISE in Arabidopsis inflorescences ». In : *The Plant Cell* 20.4, p. 888-900 (cité en page 16).
- REBETZKE, GJ, J JIMENEZ-BERNI, RA FISCHER, DM DEERY et DJ SMITH (2019). « High-throughput phenotyping to enhance the use of crop genetic resources ». In : *Plant Science* 282, p. 40-48 (cité en page 10).
- ROBERTS, Daniel A., Sho YAJIDA et Boris HANIN (mai 2022). *The Principles of Deep Learning Theory*. Cambridge University Press (cité en page 28).
- ROSENBERG, Andrew et Julia HIRSCHBERG (2007). « V-measure : A conditional entropy-based external cluster evaluation measure ». In : *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, p. 410-420 (cité en page 96).
- ROUSSEEUW, Peter J. (nov. 1987). « Silhouettes : A graphical aid to the interpretation and validation of cluster analysis ». en. In : *Journal of Computational and Applied Mathematics* 20, p. 53-65 (cité en page 57).
- RUSU, Radu Bogdan (oct. 2009). « Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments ». Thèse de doct. Computer Science department, Technische Universitaet Muenchen, Germany (cité en page 27).
- RUSU, Radu Bogdan, Nico BLODOW et Michael BEETZ (2009). « Fast point feature histograms (FPFH) for 3D registration ». In : *2009 IEEE international conference on robotics and automation*. IEEE, p. 3212-3217 (cité en page 32).

- SCHUNCK, David, Federico MAGISTRI, Radu Alexandru ROSU et al. (août 2021). « Pheno4D : A spatio-temporal dataset of maize and tomato plant point clouds for phenotyping and advanced plant analysis ». en. In : *PLOS ONE* 16.8. Publisher : Public Library of Science, e0256340 (cité en pages 32, 90, 91, 93, 100, 101, 104).
- SECORD, John et Avidesh ZAKHOR (2007). « Tree detection in urban regions using aerial lidar and image data ». In : *IEEE Geoscience and Remote Sensing Letters* 4.2, p. 196-200 (cité en page 28).
- SHUMAN, David I., Sunil K. NARANG, Pascal FROSSARD, Antonio ORTEGA et Pierre VANDER-GHEYNST (mai 2013). « The Emerging Field of Signal Processing on Graphs : Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains ». In : *IEEE Signal Processing Magazine* 30.3. arXiv : 1211.0053, p. 83-98 (cité en pages 42, 43).
- SLININGER, Brian (2013). *Fiedler's Theory of Spectral Graph Partitioning* (cité en page 49).
- SONG, Peng, Jinglu WANG, Xinyu GUO, Wanneng YANG et Chunjiang ZHAO (juin 2021). « High-throughput phenotyping : Breaking through the bottleneck in future crop breeding ». en. In : *The Crop Journal*. Rice as a model crop : genetics, genomics and breeding 9.3, p. 633-645 (cité en pages 10, 12).
- SORKINE, Olga (2005). « Laplacian mesh processing ». In : *Eurographics (State of the Art Reports)* 4 (cité en page 41).
- SPALDING, Edgar P et Nathan D MILLER (2013). « Image analysis is driving a renaissance in growth measurement ». In : *Current opinion in plant biology* 16.1, p. 100-104 (cité en page 13).
- TABB, Amy et Henry MEDEIROS (2018). « Fast and robust curve skeletonization for real-world elongated objects ». In : *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, p. 1935-1943 (cité en page 30).
- TAGLIASACCHI, Andrea, Thomas DELAME, Michela SPAGNUOLO, Nina AMENTA et Alexandru TELEA (2016). « 3d skeletons : A state-of-the-art report ». In : *Computer Graphics Forum*. T. 35. 2. Wiley Online Library, p. 573-597 (cité en page 30).
- TARDIEU, François, Llorenç CABRERA-BOSQUET, Tony PRIDMORE et Malcolm BENNETT (août 2017). « Plant Phenomics, From Sensors to Knowledge ». en. In : *Current Biology* 27.15, R770-R783 (cité en page 10).
- TATOGLU, Akin et Kishore POCHIRAJU (2012). « Point cloud segmentation with LIDAR reflection intensity behavior ». In : *2012 IEEE International Conference on Robotics and Automation*. IEEE, p. 786-790 (cité en page 26).
- THEBAULT, Ludovic (2022). *Un aperçu de botanique*. fr-FR. via <https://www.tela-botanica.org/ressources/ressources-pedagogiques/un-apercu-de-botanique/>, licence CC BY-SA 4.0 (cité en page 6).
- THORNDIKE, Robert L. (1953). « Who belongs in the family? » In : *Psychometrika* 18.4, p. 267-276 (cité en page 57).

- TURGUT, Kaya, Helin DUTAGACI, Gilles GALOPIN et David ROUSSEAU (2022). « Segmentation of structural parts of rosebush plants with 3D point-based deep learning methods ». In : *Plant Methods* 18.1, p. 1-23 (cité en pages 2, 33, 104).
- VIJAYARANGAN, Srinivasan, Paloma SODHI, Prathamesh KINI et al. (2018). « High-throughput robotic phenotyping of energy sorghum crops ». In : *Field and Service Robotics*. Springer, p. 99-113 (cité en pages 11, 32, 34).
- WAHABZADA, Mirwaes, Stefan PAULUS, Kristian KERSTING et Anne-Katrin MAHLEIN (déc. 2015). « Automated interpretation of 3D laserscanned point clouds for plant organ segmentation ». en. In : *BMC Bioinformatics* 16.1 (cité en page 32).
- WANG, Di (juil. 2020). « Unsupervised semantic and instance segmentation of forest point clouds ». en. In : *ISPRS Journal of Photogrammetry and Remote Sensing* 165, p. 86-97 (cité en pages 32, 35).
- WANG, Di, Stéphane MOMO TAKOUDJOU et Eric CASELLA (mar. 2020). « LeWoS : A universal leaf-wood classification method to facilitate the 3D modelling of large tropical trees using terrestrial LiDAR ». en. In : *Methods in Ecology and Evolution* 11.3. Sous la dir. de Ryan CHISHOLM, p. 376-389 (cité en page 34).
- WANG, Yi, Conrad M ALBRECHT, Nassim Ait Ali BRAHAM, Lichao MOU et Xiao Xiang ZHU (2022). « Self-supervised learning in remote sensing : A review ». In : *arXiv preprint arXiv :2206.13188* (cité en page 28).
- WARRENS, Matthijs J. et Hanneke van der HOEF (2022). « Understanding the Adjusted Rand Index and Other Partition Comparison Indices Based on Counting Object Pairs ». In : *Journal of Classification* (cité en page 96).
- WEINMANN, Martin, Boris JUTZI, Stefan HINZ et Clément MALLET (juil. 2015). « Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers ». en. In : *ISPRS Journal of Photogrammetry and Remote Sensing* 105, p. 286-304 (cité en pages 27, 29, 35, 37, 107).
- WIENKOOP, Stefanie, Sacha BAGINSKY et Wolfram WECKWERTH (2010). « Arabidopsis thaliana as a model organism for plant proteome research ». In : *Journal of proteomics* 73.11, p. 2239-2248 (cité en page 19).
- WINTZ, Timothée, David COLLIAUX et Peter HANAPPE (2018). « Automated extraction of phyllotactic traits from arabidopsis thaliana ». In : *CVPPP workshop, ECCV*. T. 2018, p. 6 (cité en page 17).
- WU, Sheng, Weiliang WEN, Boxiang XIAO et al. (2019). « An Accurate Skeleton Extraction Approach From 3D Point Clouds of Maize Plants ». In : *Frontiers in Plant Science* 10, p. 248 (cité en page 30).
- XIANG, Tao et Shaogang GONG (2008). « Spectral clustering with eigenvector selection ». In : *Pattern Recognition* 41.3, p. 1012-1029 (cité en page 48).

- XIE, Yuxing, Jiaojiao TIAN et Xiao Xiang ZHU (2020). « Linking Points With Labels in 3D : A Review of Point Cloud Semantic Segmentation ». In : *IEEE Geoscience and Remote Sensing Magazine* 8.4, p. 38-59 (cité en pages 25, 26, 29, 37).
- XU, Hui, Nathan GOSSETT et Baoquan CHEN (oct. 2007). « Knowledge and heuristic-based modeling of laser-scanned trees ». en. In : *ACM Transactions on Graphics* 26.4, p. 19 (cité en page 30).
- YUAN, Xiaocui, Huawei CHEN et Baoling LIU (mai 2020). « Point cloud clustering and outlier detection based on spatial neighbor connected region labeling ». en. In : *Measurement and Control*, p. 002029402091986 (cité en page 28).
- ZHAO, Chunjiang, Ying ZHANG, Jianjun DU et al. (2019). « Crop Phenomics : Current Status and Perspectives ». In : *Frontiers in Plant Science* 10, p. 714 (cité en pages 10, 11).
- ZIAMTSOV, Illia et Saket NAVLAKHA (déc. 2019). « Machine Learning Approaches to Improve Three Basic Plant Phenotyping Tasks Using Three-Dimensional Point Clouds ». en. In : *Plant Physiology* 181.4, p. 1425-1440 (cité en pages 30-32).

Table des figures

2.1	Schéma d'une feuille simple pétiolée (Extrait de THEBAULT, 2022) . . .	6
2.2	Quelques différentes formes de feuilles et les différents types de bordures. Extrait de FORTIER, 2022 disponible à l'adresse https://www.aquaportail.com/definition-3775-feuille.html	7
2.3	Le processus de pousse d'une plante. Extrait de BARTHÉLÉMY et Yves CARAGLIO, 2007. (A) On voit sur ce schéma un méristème protégé par un bourgeon. Des feuilles sont générées. La cataphylle est une feuille dont l'objectif n'est pas la photosynthèse mais est utilisée pour une autre fonction dans la plante. Dans le cas présent, la cataphylle protège le bourgeon. (B) Ce méristème est désigné comme apical car sur la partie la plus haute de la plante. Il contribue donc à sa croissance en hauteur. Les notions de nœuds et entre nœuds sont également illustrées sur la tige de la plante.	8
2.4	Les différentes dispositions des feuilles sur une tige (BELL, 1991). (a) monostique (b) monostique spiralée, (c) distique, (d) distique spiralée, (e) tristique, (f) tristique spiralée, (h) alterne spiralée ou hélicoïdale, (i) opposé, (j) opposé décussé, (k) opposé spiralé, (l) verticillée). Extrait de BELL, 1991.	9
2.5	Schéma des interactions génotype-phénotype-environnement et des différentes échelles d'observation possibles de la plante (Extrait de ZHAO et al., 2019)	11
2.6	Les différents environnements de phénotypage et d'acquisition des données. Extrait de SONG et al., 2021. L'environnement de phénotypage doit s'adapter selon les traits mesurés, les traits microscopiques, à l'échelle des cellules seront observés dans un laboratoire. Les traits macroscopiques peut être mesurés dans une serre, un champ expérimental ou un champ de cultures.	12
2.7	Bilan des méthodes d'acquisition disponibles pour obtenir un modèle 3D de plante adapté au phénotypage de traits caractéristiques macroscopiques telles que la surface foliaire, les distances entre nœuds ou encore la hauteur de tige.	14

2.8	Plateforme de l'imageur ROMI. Le scanner est composé d'un cadre en aluminium, d'un CNC combiné à un support inclinable et d'un appareil photo. Extraits de https://romi-project.eu/	17
2.9	Spécimen de chénope pour tests. La photographie représente la plante dans le robot au moment de l'acquisition. C'est une des images utilisées pour la reconstruction du modèle 3D présenté à droite. Le nuage de points est coloré de façon aléatoire en fonction de l'indexe des points pour obtenir un résultat plus lisible visuellement. Ce nuage est constitué de 55 918 points.	18
2.10	Spécimen de chénope virtuel pour tests. L'image de gauche présente la plante générée dans L-Py. A droite le nuage de points généré à partir de ce modèle. Il est possible de transférer des informations du modèle sur les points du nuage. Les éléments sont donc ici colorés en fonction de trois classes sémantiques, en vert les limbes, en bleu les pétioles et branches, enfin en rouge la tige principale. Ce nuage de points issu d'une plante virtuelle est constitué de 10 000 points.	18
2.11	A gauche, une photographie d'Arabidopsis utilisée pour la reconstruction du modèle 3D présenté à sa droite. Le nuage de points obtenu est à nouveau coloré grâce aux numéro de points pour plus de lisibilité. Ce nuage de points est constitué de 4440 points. Enfin à droite, un nuage de points d'Arabidopsis obtenu à partir d'un modèle virtuel est présenté. Il est constitué de 8190 points. On a également pu obtenir une classification sémantique en trois classes, bleue la tige principale, rouge les pétioles, vert les siliques.	19
3.1	Chaîne de traitement générale pour une segmentation de données. Extrait de JAIN et al., 1999.	24
3.2	Les grandes familles de méthodes de segmentation par instance des nuages de points selon et repris de NGUYEN et LE, 2013.	25
3.3	Chaîne de traitement pour classifier un nuage de points avec une méthode de classification supervisée extraite de WEINMANN et al., 2015.	27
3.4	Chaîne de traitement générale pour la reconstitution d'un modèle de plante à partir de nuages de points. Cette chaîne de traitement inclut la construction d'un squelette. A partir de CHAUDHURY et Christophe GODIN, 2020a.	30
3.5	Exemple de squelettes superposés aux nuages de points de deux plantes à différents moment de croissance. Extrait de CHEBROLU et al., 2021.	31

3.6	Exemple d'erreurs rencontrées avec l'utilisation d'un réseau neuronal (ici PointCNN). Extrait de TURGUT et al., 2022. La segmentation sémantique comprend trois classes : les tiges/pétiotes/branches en rouge, les fleurs en bleu et les limbes en vert.	33
4.1	Exemple de graphe non orienté et avec une pondération constante égale à 1 avec 4 sommets et sa matrice d'adjacence associée.	39
4.2	Illustration d'un signal de graphe en chaque noeud du graphe de Petersen. La hauteur de chaque barre bleue représente la valeur du signal au noeud sur lequel repose la barre. Extrait de SHUMAN et al., 2013	42
4.4	Une matrice d'incidence orientée correspondant au graphe exemple en figure 4.1	43
4.5	Étapes de calcul pour obtenir la décomposition spectrale du nuage de points et l'utiliser pour segmenter le nuage. Les modules Python utilisés pour le calcul sont affichés en rouge.	44
4.6	Temps de calcul de la matrice laplacienne en fonction du nombre de plus proche voisins par point	45
4.7	Temps de calcul des vecteurs propres en fonction de la taille de la matrice laplacienne, autrement dit du nombre de points contenus dans le nuage.	45
4.8	Décomposition dans le domaine spectral d'un nuage de point d'arabette type "arête de poisson". Seuls les vecteurs propres 2, 3, 4, 18 et 25 ont été représentés ici. Graphes du haut : les valeurs des composants de chaque vecteur propre représenté en fonction des noeuds (en x). Les noeuds sont triés une première fois pour obtenir une courbe croissante des composantes lors de l'affichage du vecteur propre 2, puis cet ordre des noeuds est gardé pour les courbes suivantes. Graphes du bas : Les nuages de points sont colorés en fonction des valeurs de chaque vecteur propre. Les plus petites valeurs sont de couleur bleue et les plus grandes de couleur rouge.	46
4.9	Impact du choix du nombre de vecteurs propres sur une segmentation effectuée grâce à la méthode de partitionnement spectrale décrite dans LUXBURG, 2007. K donne le nombre de vecteurs propres utilisés, c le nombre de parties demandées en entrée de l'algorithme des K-means.	47
4.10	Impact du choix du nombre de clusters demandés en entrée de l'algorithme des K-means sur une segmentation effectuée grâce à la méthode de partitionnement spectrale de LUXBURG, 2007. K donne le nombre de vecteurs propres utilisés, c le nombre d'instances demandées en entrée de l'algorithme des K-means.	48

4.11	Comportement du vecteur de Fiedler sur un exemple de graphe de la forme d'une chaîne élémentaire simple. a) Les différents nœuds du graphe sont représentés par des points colorés. Les arêtes du graphe ne sont pas affichées sur cette illustration. Néanmoins, chaque nœud est relié par une arête aux deux nœuds les plus proches. L'axe principal du graphe est composé de 100 nœuds. Les deux axes secondaires sont chacun composés de 20 nœuds. A chaque nœud a été attribué une couleur en fonction de la valeur que prend le vecteur de Fiedler en ce point. Les valeurs du vecteur de Fiedler aux nœuds aux extrémités du graphe ont été affichés. Le sens de variation positive des valeurs est indiqué par des flèches. b) Ce graphe représente les valeurs du vecteur de Fiedler en fonction des nœuds classés de haut en bas du graphe représenté en a) sans les axes secondaires. c) Le vecteur de Fiedler a été calculé sur le graphe a) entier, avec les axes secondaires.	50
4.12	Valeurs du vecteur de Fiedler en fonction des nœuds du graphe présenté en figure 4.11 a). Les nœuds des axes latéraux ont été isolés et leurs valeurs affichées à la fin de la courbe, à partir de l'index 100.	51
4.13	Vecteurs de Fiedler calculés sur deux nuages de points différents. Les valeurs du vecteur de Fiedler en chaque point sont représentées grâce à une échelle de couleur "Bleu->Vert->Jaune->Rouge". Quelques valeurs en certains points d'extrémités sont affichées sur les figures et les sens d'évolution des valeurs sont indiquées par des flèches.	53
4.14	Champ de vecteurs normé calculé à partir du gradient du vecteur de Fiedler.	54
4.15	Champ de vecteurs calculé à partir du gradient du vecteur de Fiedler. La taille des traits est proportionnelle à la norme de chaque vecteur. . . .	54
4.16	Nuage de points de chénopode réel issu de la chaîne de traitement ROMI. Chaque point est coloré selon la valeur de la norme du gradient du vecteur de Fiedler associé. L'échelle de couleur utilisée est "Bleu->Vert->Jaune->Rouge".	55
4.17	Résultats suite au calcul du gradient du vecteur de Fiedler et à l'exploitation de sa norme par l'application de l'algorithme des K-means avec K=4. La partie bleue correspond à la classe rassemblant les points portant les normes les plus petites. Il correspond donc aux limbes et apex.	58

4.18	Résultat de la segmentation par instance. Le nuage de points a été pré-segmenté grâce au K-means sur la norme du GDM de Fiedler puis a subi un traitement de croissance de région. Il est donc segmenté par instance, chaque couleur représente une instance. On notera que chaque limbe est maintenant dans sa propre instance. L'algorithme a donc accompli sa mission. Néanmoins, on note également la présence de défauts fréquents suite à l'utilisation d'un algorithme ne prenant pas directement en compte la topologie du graphe de similarité construit ou les coordonnées des points du nuage. Les cercles verts montrent des instances isolées et mal segmentées. On peut voir dans le cercle rouge deux instances mélangées entre elles. Enfin, le cercle violet montre des frontières mal définies entre les instances présentes au bas de la feuille de ce modèle de chénopode.	61
4.19	Différents cas de figure pour le calcul de l'énergie topologique en un nœud v , valeur de l'énergie avec la formule intuitive $\mathcal{E}_{int}(v)$ et la formule finale utilisée $\mathcal{E}(v)$	64
4.20	Défauts de segmentation résolus suite à l'application du filtre basés sur l'énergie topologique en chaque point.	65
5.1	a)-b-c) Schéma de construction d'un graphe quotient. La première étape Le nuage de points segmenté par instance (de façon encore imparfaite) associé est avec le même angle de vue est affiché en dessous. La deuxième étape est représentée par un schéma reprenant une partie de la structure de graphe de similarité qui aurait été segmenté en 4 parties. Le graphe quotient est représenté dans la troisième partie. Chaque partie connexe du graphe de similarité est représentée par un nœud dans le graphe quotient. d) Représente le graphe de similarité sur une plante réelle (<i>chenopodium album</i>). Un zoom a été effectué pour permettre de constater de la structure de graphe de l'objet. e) Segmentation par instance sur un plant de chénopode f) Superposition d'un nuage de points de chénopode et de son graphe quotient associé, les points bleus appartiennent à la classe limbe/apex, les autres couleurs sont pour les parties linéaires non classés. Les points du graphe quotient ont été placés en 3D en calculant les coordonnées moyennes de l'ensemble des points de la partie qu'ils représentent.	69
5.2	Schéma descriptif de la méthode de division/fusion	72

5.3	Action de l'algorithme de division/fusion sur la segmentation du nuage de points. Le nuage de points utilisé ici est un nuage de points de chénopode réel. Les points sont colorés grâce à la segmentation par instance réalisée. Chaque couleur représente une instance.	73
5.4	Extrema locaux des valeurs du vecteur de Fiedler sur trois organes différents. a) Un limbe b) un apex c) un limbe dentelé.	76
5.5	Détermination de la tige principale avec le graphe quotient. Le nœud racine est présenté en rouge. Les nœuds représentant les organes linéaires indifférenciés sont en bleu clair au début de l'algorithme. Les nœuds représentant des apex ou des limbes sont en bleu foncé. a)-c) Les plus courts chemins de chaque limbe ou apex jusqu'au nœud racine sont calculés et une charge est ajoutée à chaque nœud pour chacun des chemins qui l'emprunte. d) Le nuage de points présente une segmentation, chaque point a été coloré en fonction de la charge calculée pour son cluster. L'échelle de couleur va du bleu (minimum) vers le rouge (maximum). .	77
5.6	Évolution du graphe quotient vers un arbre sémantique au fur et à mesure des algorithmes. Chaque couleur correspond à un organe, en bleu les limbes, jaune les apex, orange les pétioles, rouge les branches et vert la tige principale.	82
5.7	Changements apportés par l'étape d'amélioration de la segmentation sur un chénopode réel. La segmentation présentée est une segmentation sémantique, chaque point du nuage a la couleur de la classe à laquelle il appartient. Les limbes sont en bleue, les apex en jaune, la tige principale en vert, les branches en rouge et les pétioles en orange. Une erreur de sous-segmentation d'un apex a été identifiée et entourée en rouge. Un cotylédon est entouré en noir. a) Segmentation sémantique sans étape d'amélioration b) Segmentation sémantique après étape d'amélioration.	84
6.1	Chaîne de traitement mise en place pour la segmentation de nuages de points de plantes.	88
6.2	Temps de calcul de l'ensemble de la chaîne de traitement en fonction du nombre de points présents dans le nuage. La première mesure a été effectuée sur un chénopode virtuel, les deux suivantes sur des chénopodes réels et les trois dernières sur des plants de tomates issus de Pheno4D. Les plantes ont été sélectionnées de façon à présenter des difficultés et un nombre de parties à atteindre relativement égales.	90

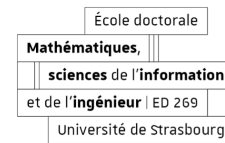
6.3	Jeux de données utilisés pour l'évaluation du pipeline. Photographies ou captures d'écran des objets de a) à c) : a) chénopode réel, b) chénopode virtuel, c) plant de tomates (photo d'illustration, indépendante du jeu de données de SCHUNCK et al., 2021. Les nuages de points correspondants sont présentés de d) à f). On notera que d) et e) ont les mêmes angles de vues et représentent exactement les objets présentés en a) et b).	91
6.4	Résultats visuels de la segmentation sémantique réalisée par la chaîne de traitement. Les points sont colorés en fonction de la classe sémantique qui leur est attribuée, bleu pour les limbes, vert pour la tige principale, orange pour les pétioles, rouge pour les branches et jaune pour les apex. a-b) Segmentation sémantique de deux chénopodes virtuels différents. c-d) Même segmentations sémantiques que les deux chénopodes présentés en a-b) mais les classes apex ont été fusionnées avec les limbes. e-f) Segmentation sémantique de deux chénopodes réels. g-h) Segmentation sémantique de deux plants de tomate réels.	94
6.5	Défauts de la segmentation (a, c, e) et impact de l'étape d'amélioration (b, d, f). Les limbes sont en bleu, les tiges en vert et les pétioles en orange. Les figures a) et b) présentent la difficulté d'obtenir des frontières nettes lorsque le pétiole est très petit. Les figures e) et f) présentent une erreur de sous-segmentation d'un limbe et d'un pétiole, erreur corrigée grâce à l'étape d'amélioration.	98
6.6	Segmentations de références (a, c) et segmentations obtenues grâce à la chaîne de traitement proposée (b, d). Les limbes sont en bleu, les tiges en vert et les pétioles en orange. Dans le premier exemple, la segmentation de référence (a) est moins segmentée que la segmentation obtenue (b). La segmentation issue de la chaîne de traitement présente un niveau de détail plus important. Dans le second exemple, les deux branches ont été détectées dans le résultat (d) mais n'englobent pas suffisamment de points du nuage par rapport à la référence (c).	99
6.7	F_1 -score sur la segmentation de deux plants de tomates différents à 7 stades de croissance différents.	101

Liste des tableaux

2.1	Tableau décrivant certaines des caractéristiques de l'Arabidopsis thaliana et du Chenopodium album à l'âge adulte. Données synthétisées à partir de INRA, 2000. Les informations sur les organes qui ne nous intéressent pas dans cette étude ont été grisées.	20
6.1	Résultats de la segmentation sémantique (en %) sur le jeu de données des chénopodes virtuels, avec et sans l'étape d'amélioration (section 5.2.5)	97
6.2	Résultats moyens (en %) de la segmentation sémantique sur le jeu de données des 5 chénopodes réels. Les résultats sont présentés sans et avec l'étape d'amélioration expliquée en section 5.2.5.	98
6.3	Résultats moyens (en %) de la segmentation sémantique en deux classes effectuée sur 14 plants de tomates. Résultats de la segmentation sémantique obtenue par SCHUNCK et al., 2021 avec trois réseaux neuronaux différents et sur le jeu de données de Phéno4D entier.	101
6.4	Moyenne des scores obtenus pour chaque segmentation par instance calculée pour les 5 chénopodes virtuels traités. Les scores ont été calculés avec et sans l'étape d'amélioration de la segmentation expliquée en section 5.2.5.	102
6.5	Comparaison des surfaces de limbes totales et des indices de surface foliaire entre la segmentation en sortie de la chaîne de traitement et la segmentation de référence. Comparaison de la hauteur de tige principale obtenue à partir de la segmentation de référence et de la segmentation résultat de la chaîne de traitement.	103

Katia Mirande

Segmentation sémantique et par instance de nuages de points 3D de plantes



Résumé

L'obtention simultanée d'une segmentation sémantique et par instance de nuages de points 3D de plantes est une étape indispensable pour le phénotypage automatique de plantes. Chaque organe de la plante est généralement détecté à partir de la géométrie locale du nuage de points, mais la cohérence globale de la structure de la plante est rarement utilisée.

Dans cette thèse, nous explorons l'utilisation de deux graphes à échelles différentes pour segmenter et classifier des nuages de points de plantes. A l'échelle du point, le graphe de similarité permet de calculer des attributs géométriques basés sur le spectre du graphe et de distinguer les organes linéaires (tige principale, branches, pétioles) des organes à deux dimensions et plus (limbes et apex). A l'échelle des organes, le graphe quotient est utilisé pour obtenir une classification détaillée et corriger des défauts potentiels.

Nous proposons ainsi une chaîne de traitement rapide, automatique et intégrant la structure de la plante. Nous évaluons cette dernière sur des bases de données de nuages de points 3D de *Chenopodium album* (chénopodes blanc) et de *Solanum lycopersicum* (plants de tomates).

Mots-clés : segmentation par instance, segmentation sémantique, vecteur de Fiedler, graphe quotient, phénotypage

Résumé en anglais

Accurate simultaneous semantic and instance segmentation of a plant 3D point cloud is critical for automatic plant phenotyping. Classically, each organ of the plant is detected based on the local geometry of the point cloud, but the consistency of the global structure of the plant is rarely assessed.

In this thesis, we explore the use of two graphs at different scales to segment and classify 3D plant point clouds. At the point scale, the similarity graph allow us to compute geometrical attributes from the spectrum of the graph and to distinguish between linear organs (main stem, branches, petioles) and two or three-dimensional ones (leaf blades, apices). At the organ scale, the quotient graph is used to obtain a detailed classification and correct potential segmentation defects.

We propose a fast and automatic pipeline that integrates knowledge of the structure of the plant. The method is assessed on both synthetic and real 3D point-cloud data sets of *Chenopodium album* (wild spinach) and *Solanum lycopersicum* (tomato plant).

Keywords : instance segmentation, semantic segmentation, Fiedler vector, quotient graph, phenotyping