



Mathématiques Appliquées et Modélisation

Rapport de stage

Sujet du stage :

Reconstruction spatio-temporelle à haut débit de l'architecture du système racinaire à partir de l'imagerie frugale en utilisant le suivi topologique et l'apprentissage profond.

Étudiant	Loïc Gandeel (MAM5 – SCIENCE DES DONNÉES)
Maître de stage	Romain Fernandez
Tuteur Enseignant	Lionel Fillatre
Organisme de recherche	CIRAD (Montpellier)
Dates du stage	04/03/2024 - 03/09/2024

Table des matières

1	Introduction	3
2	Description du travail proposé	5
2.1	Contexte	5
2.1.1	Architecture des systèmes racinaires	5
2.1.2	Plateformes de phénotypage des racines	6
2.1.3	Reconstruction de modèles numériques de systèmes racinaires	7
2.1.4	Limitations actuelles des méthodes de reconstruction de l'architecture racinaire	9
2.2	Problématique générale du stage	11
2.3	Objectifs du Stage	11
3	Description du travail réalisé	13
3.1	Vue d'ensemble des actions menées et planning du stage	13
3.2	Évolution du format RSML	13
3.2.1	Cadre théorique et algorithmes de conversion	14
3.2.2	Outil logiciel pour les annotations de RSML temporels : RSML_Expert	23
3.3	Préparations d'un data challenge d'architecture spatio-temporelle	24
3.3.1	Description du challenge	25
3.3.2	Constitution du jeu de donnée	25
3.3.3	Description des étapes de préparation des données	26
3.3.4	Plan de gestion des données - FAIR	27
3.4	Optimisations et améliorations des outils de recalage d'images	28
4	Conclusion et perspectives	30

Remerciements

Je tiens à exprimer ma profonde gratitude à mon maître de stage, Romain Fernandez. Ses conseils, son soutien et ses encouragements constants ont été déterminants tout au long de ce stage. Son expertise et sa perspicacité ont été déterminantes pour me permettre de mieux appréhender le sujet et la complexité du projet.

Je suis également très reconnaissant envers Christophe Pradal pour son mentorat et pour m'avoir offert l'opportunité de travailler sur ce projet passionnant. Ses conseils et son soutien ont grandement contribué à mes progrès et à mon développement professionnel.

Je tiens à adresser mes sincères remerciements à tous les membres du consortium : Reza Akbarinia, Philippe Nacry, Amandine Crabos, Cannelle Armengaud, Zheng Wingyu, Sandrine Ruffel, Bertrand Muller, Denis Fabre, Christophe Jourdan, Thomas Laisne, Raoul Vial, Valentin Aillery, Marion Prudent, Tristan Gerault, Maria Camila Rebolledo et Frederic Rees. Leur collaboration et leur assistance ont permis de rassembler un ensemble de données diversifié et complet, sans lequel ce projet n'aurait pas été possible.

Je voudrais également remercier toute l'équipe de Phenomen pour son soutien et ses encouragements. Leurs commentaires constructifs et suggestions pertinentes ont été essentiels pour façonner mon travail.

Je tiens à exprimer ma gratitude à [#DigitAg](#) pour avoir financé cette bourse et permis la réalisation de ce stage.

Mes sincères considérations s'adressent à l'équipe académique de Polytech Nice qui m'a permis de préparer ce stage dans les meilleures conditions.

Enfin, je remercie chaleureusement ma famille et mes amis pour leurs encouragements et leur compréhension tout au long de ce parcours. Leur soutien amical a été une source de force et de motivation.

Supervision

- **Romain Fernandez**, Chercheur en analyse d'images. [Profile Google Scholar](#)
romain.fernandez@cirad.fr
- **Christophe Pradal**, Chercheur en Informatique. [Profile Google Scholar](#)
christophe.pradal@cirad.fr

Introduction

Mon stage s'est déroulé au sein du département BIOS du [CIRAD](#)¹ à Montpellier, au sein de l'unité mixte de recherche [AGAP](#)², spécialisée dans la recherche en génétique et en amélioration des plantes, avec un accent particulier sur l'adaptation des plantes aux changements climatiques. J'ai travaillé dans l'équipe [PhenoMen](#)³, une équipe interdisciplinaire comprenant 18 chercheurs permanents rassemblés par axes thématiques : *Écophysio*logie, *Modélisation des plantes au Sud* et *Mathématiques et informatiques pour la modélisation*, axe auquel j'étais rattaché.

L'équipe Phenomen développe des méthodes de phénotypage pour mieux comprendre l'adaptation des plantes aux variations environnementales, et en particulier le rôle du système racinaire. Les racines sont essentielles pour l'absorption des nutriments et de l'eau, la fixation du carbone atmosphérique et l'interaction avec le sol, influençant ainsi l'efficacité de l'utilisation des ressources et la résilience des cultures face aux stress environnementaux. Le phénotypage du système racinaire permet ainsi d'étudier l'adaptation des plantes, leur nutrition, leur développement dynamique, la complémentarité des espèces en agroforesterie et la capacité de rhizodéposition du carbone atmosphérique pour l'atténuation du changement climatique.

Pour observer les racines, des plateformes de phénotypage (mesure de traits des spécimens) ont été développées, initialement pour mesurer des traits généraux tels que la largeur et la profondeur des racines, à partir d'images des systèmes racinaires. Avec le temps, ces plateformes ont évolué pour analyser en laboratoire des traits plus précis liés au développement et à la réponse dynamique des plantes. Cependant, ces expériences, réalisées en conditions contrôlées, ne suffisent pas pour comprendre les interactions racinaires en conditions réelles. Il est donc nécessaire de diversifier les moyens d'observation et de passer du laboratoire au champ.

Mon stage s'inscrit dans un projet visant à développer le phénotypage spatio-temporel des architectures racinaires des plantes à partir de données hétérogènes d'imagerie. Les résultats attendus sont la possibilité de capturer automatiquement la topologie et la géométrie des racines au fil du temps, malgré les occlusions (croisements entre racines) ou la variabilité des conditions d'observation (différents appareillages d'acquisition). Actuellement, les méthodes d'analyse automatique de ces données de phénotypage ne fournissent pas de résultats fiables sur la reconstruction architecturale ni sur la dynamique du développement racinaire en conditions de culture au champ ou en peuplements mixtes. Pour surmonter ces limitations, le projet RootSystemTracker utilise des développements récents en analyse spatio-temporelle de l'architecture racinaire ([1, Fernandez et al., 2022]).

Durant ce stage, j'ai bénéficié du financement de [#DigitAg](#), un institut de convergence de l'Université de Montpellier qui vise à promouvoir la recherche interdisciplinaire en agriculture numérique. Mon objectif principal au cours de ce stage était de contribuer à l'amélioration des techniques de reconstruction de l'architecture racinaire en travaillant sur l'optimisation des algorithmes de reconstruction, sur l'extension de standards d'annotations de données pour inclure des informations temporelles dynamiques, puis en me consacrant à la construction d'un jeu de données issues de différentes plateformes de phénotypage racinaire pour fournir des données d'entraînement et de validation pour la suite du projet, et le diffuser dans la communauté comme un data challenge.

En cours de stage, j'ai également candidaté et obtenu un [financement du PEPR Agroécologie et Numérique](#) pour une thèse qui débutera en octobre 2024. L'objectif consistera à s'appuyer sur les données du data challenge

1. CIRAD : Centre de coopération Internationale en Recherche Agronomique pour le Développement

2. AGAP : Amélioration Génétique et Adaptation des Plantes méditerranéennes et tropicales

3. PhenoMen : Phénotypage et Modélisation des plantes dans leur environnement agro-climatique

pour améliorer les modèles et algorithmes de reconstruction spatio-temporelle, et concevoir des outils de Data Mining permettant de détecter des similarités et/ou des anomalies dans les patterns de croissance des plantes.

Annonce du plan

Dans une première partie, je dresserai d'abord un panorama de l'environnement dans lequel s'inscrit ce projet. Plus précisément, je traiterai des éléments suivants :

- **i) Le contexte du stage**, où je mettrai en lumière des concepts tels que l'architecture des systèmes racinaires, l'importance du phénotypage dans le domaine agricole, ainsi qu'une revue des plateformes de phénotypage existantes.
- **ii) Les objectifs du projet RootSystemTracker**, en présentant l'élaboration de nouvelles méthodes destinées à l'analyse spatio-temporelle des systèmes racinaires.

Dans une deuxième partie, je me concentrerai sur les réalisations et l'avancement du projet durant la période de stage. Cette partie se subdivisera en plusieurs sous-sections :

- **i) Les actions menées et le planning du stage**, où je détaillerai les tâches spécifiques qui ont été accomplies et les différentes phases du développement.
- **ii) L'évolution du standard RSML vers RSML $2d + t$** , en expliquant comment ce standard a été enrichi avec l'intégration de la dimension temporelle.
- **iii) Constitution d'un jeu de données**, où je décrirai les différentes étapes de collecte, de nettoyage et de structuration des données, ainsi que les outils et les méthodes utilisés pour ce faire.
- **iv) Les optimisations et développements logiciels**, où je discuterai des améliorations apportées aux outils informatiques utilisés, ainsi que des perspectives pour la poursuite du projet au niveau doctoral.

Description du travail proposé

2.1 Contexte

Observer et analyser les structures souterraines des plantes représente un défi majeur en raison de leur inaccessibilité et de leur complexité géométrique et topologique (ordre des ramifications en particulier). Cette difficulté est exacerbée par la nécessité d'étudier un large éventail de variétés de plantes afin de mieux comprendre leurs mécanismes d'adaptation aux stress environnementaux et de permettre le développement de variétés plus résilientes et productives. Le **phénotypage des racines**, c'est-à-dire la caractérisation de l'ensemble des traits observables des racines des plantes, est donc un enjeu majeur pour la recherche agronomique et agro-écologique, pour aux objectifs précédemment cités. Cela passe par l'étude de l'architecture racinaire des plantes dans l'espace et dans le temps.

2.1.1 Architecture des systèmes racinaires

L'architecture racinaire désigne la structure des racines d'une plante, qui se compose de différents types de racines (primaires, séminales, latérales ou adventives). Cette architecture est influencée par des signaux environnementaux et génétiques et joue un rôle clé dans l'adaptation des plantes aux conditions changeantes de leur environnement.

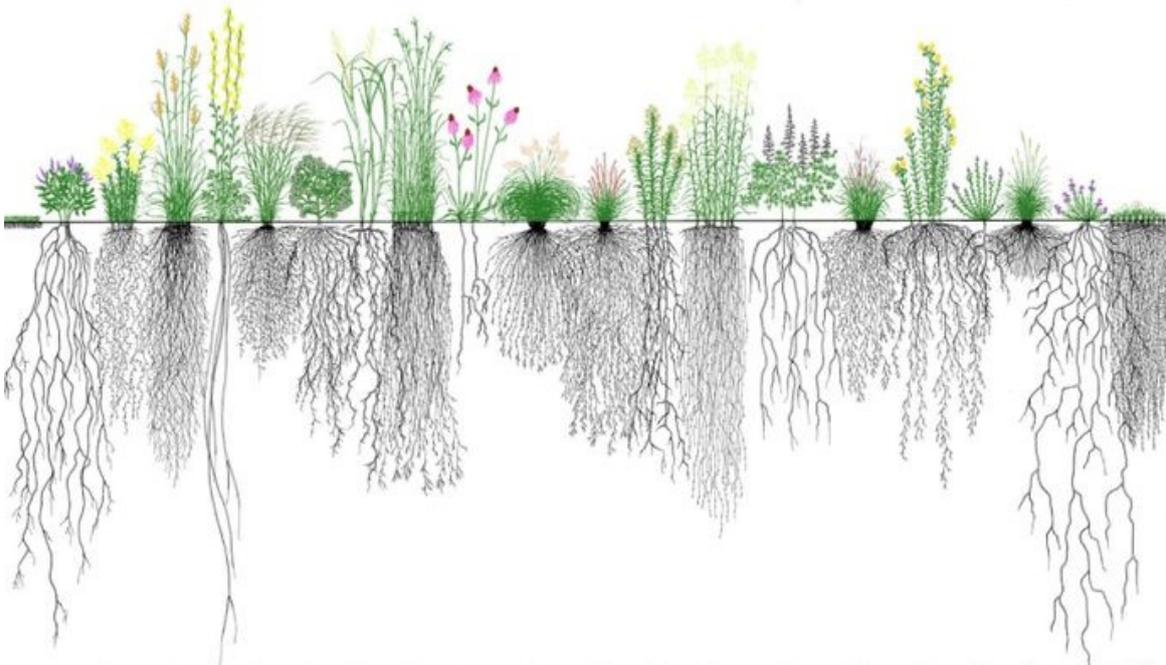


FIGURE 2.1 – [Illustration](#) de la diversité des architectures racinaires

Phénotypage des racines et importance pour l'agriculture

Le phénotypage racinaire permet de mesurer et d'analyser les caractéristiques des racines des plantes, telles que leur longueur, leur diamètre, leur profondeur, leur ramification, etc. de la façon la plus automatique possible. Il permet ainsi de comprendre les mécanismes d'adaptation des plantes aux stress environnementaux. Une méthode courante est de réaliser du phénotypage de populations de plantes, et d'observer le développement de plusieurs spécimens cultivés dans des conditions expérimentales comparables. L'observation du comportement des populations de plantes et l'étude du développement de leurs racines permettent de développer des variétés plus résilientes et productives, car la structure du système racinaire influence directement l'efficacité de la plante pour capturer de l'eau et des nutriments dans le sol, ainsi que la santé globale de la plante. L'exploration approfondie du lien entre l'architecture racinaire, sa dynamique, la physiologie végétale et les traits agronomiques est donc cruciale pour la sélection végétale. Le phénotypage racinaire permet également de caractériser des interactions inter-spécimens telles que la complémentarité des espèces en agro-foresterie, la compétition pour les ressources entre spécimens et la symbiose avec les micro-organismes du sol.

2.1.2 Plateformes de phénotypage des racines

Afin de caractériser l'architecture racinaire de nombreuses variétés d'une même espèce de plantes (ou génotypes), des plateformes de phénotypage ont été développées pour observer et mesurer les systèmes racinaires en croissance. Ces plateformes permettent de collecter des images de plantes de différents génotypes (G) poussant dans différentes conditions environnementales (E), donnant lieu à des études d'interactions entre les génotypes et les environnements, appelées (GxE). Par exemple, la plateforme [HIRROS](#)¹ capture automatiquement 150 images de systèmes racinaires (~ ×5) en croissance issus d'*Arabidopsis thaliana*, de tomate, de médicago ou de riz, avec une résolution d'image de 20 µm, à une fréquence d'acquisition allant de 1 à 12 images par jour.

Il existe plusieurs types de plateformes de phénotypage racinaire non-destructives, qui se distinguent par leur débit, la qualité des images qu'elles peuvent produire (résolution, rapport signal sur bruit, structure des parasites), leur coût et leur degré de destruction : destructives ou non-destructives. Un avantage significatif des plateformes de phénotypage non-destructives est qu'elles permettent de suivre, de leurs naissances à leurs morts, des segments de racines spécifiques à intervalles de temps réguliers sans avoir d'impact significatif sur les mécanismes racinaires *in situ* [2] (car on ne retire pas la racine du sol), ce qui permet d'observer la croissance progressive, les motifs de croissance et d'autres fonctions souterraines des plantes, telles que la déposition et la consommation de l'eau.

Les plateformes à haut débit² permettent de collecter des images de plantes en croissance à un rythme soutenu et dans des conditions contrôlées, tandis que les plateformes à bas débit³ permettent de collecter des images de plantes à un rythme plus lent, mais dans des conditions plus proches de leur environnement naturel.

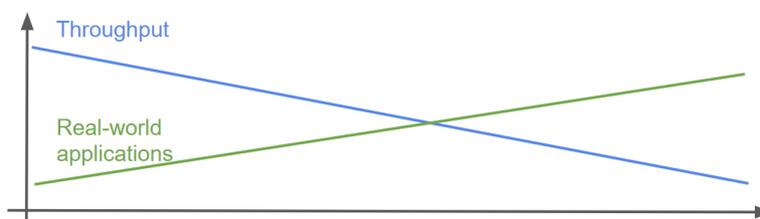


FIGURE 2.2 – Vision synthétique du compromis réalisé lors du choix du système d'acquisition des images, compromis entre la possibilité d'observer de nombreux spécimens (courbe bleue) et la possibilité de tirer des conclusions applicables à des conditions réelles de culture (courbe verte).

1. High Resolution Root Scanner
2. High ThroughPut : HTP
3. Low ThroughPut : LTP

Ces dernières années, la communauté a pu observer que les conclusions agronomiques extraites des expériences de phénotypage peuvent être plus ou moins généralisables aux conditions réelles de culture [3]. En particulier, les plateformes de phénotypage à haut débit, telles que les robots imageurs, minimisent souvent le maintien de conditions de cultures “plausibles” (culture dans la gélose par exemple). En revanche, les plateformes dites moyen ou bas débit, comme les Rhizotrons, les Rhizoscopes, les Rhizotubes ou les scanners, permettent des observations dans des conditions de culture plus proches du champ.

TABLE 2.1 – Caractéristiques des différents systèmes d’acquisition de données pour l’analyse de l’architecture racinaire

Système d’Acquisition	Rhizotrons	Rhizoscopes	Rhizotubes	Robots imageurs (HIR-ROS, AgarBox, Buvard-Box)	Scanners
Débit	Bas débit	Moyen débit	Moyen débit	Haut débit	Bas débit
Brève Description	Dispositif d’acquisition non-destructif consistant en une boîte transparente insérée autour duquel les racines peuvent se développer et dans lequel on insère des caméras pour capturer des images des racines.	Outil de phénotypage racinaire en condition semi-hydroponique, permettant une évaluation fine des racines en utilisant des billes de verre comme substrat.	Dispositif cylindrique (RhizoTubes) permettant la croissance de 1 à 6 plantes simultanément, avec une hauteur maximale de 1,1 m et jusqu’à 8 semaines, selon l’espèce végétale. Imagerie automatique et non destructive du compartiment racinaire à l’aide de l’unité d’imagerie (RhizoCab).	Automate de prise de vue de racines de plantes cultivées en boîte de Pétri. Acquisition automatique et périodique d’images pour l’analyse de l’architecture du système racinaire de plantules de diverses espèces.	Utilisation de scanners pour l’imagerie des systèmes racinaires sur de grands volumes de sol.
Gestion Autonome	Non	Partielle. Jusqu’à 192 plaques mesures 50×20×2cm.	Paritelle. Une plante par tube.	Oui. Jusqu’à 150 boîtes 12×12 cm ou 72 boîtes 24×24 cm.	Non
Intervalles d’Acquisition	Variable entre quelques jours et quelques mois.	Variable entre quelques minutes et plusieurs jours.	Variable.	Intervalles d’acquisition allant de 2h à 24h.	Variable
Espèces pouvant être étudiées	Multiples	Riz, blé, orge, sorgho, coton, eucalyptus, pois chiche...	Medicago truncatula, Pisum sativum, Brassica napus, Vitis vinifera, Triticum aestivum, Vulpia myuros, etc.	Arabidopsis thaliana, tomate, médicago, riz...	Multiples

Dans le cadre de mon stage, nous avons collecté et mis en forme des données hétérogènes provenant de ces diverses (cf. figure 2.3). L’idée est de pouvoir répondre favorablement aux différents “points” du compromis entre le débit et la plausibilité, et de faciliter les étapes de la construction scientifique en biologie, depuis le haut débit vers le bas débit. L’équipe avec laquelle j’ai travaillé a ainsi recueilli des données d’images de systèmes racinaires provenant de plusieurs dispositifs d’acquisition, dans des conditions d’acquisition différentes, avec des espèces différentes, et en cas de peuplements mixtes ou de plantes isolées, etc. J’ai par la suite travaillé sur le développement d’un outil permettant le phénotypage automatique de systèmes racinaires, adapté aux différents systèmes d’acquisition, aux différentes espèces, aux peuplements mixtes, et notamment utilisable pour des données récoltées in situ.

2.1.3 Reconstruction de modèles numériques de systèmes racinaires

Les ensembles d’images produites peuvent être utilisés, par exemple, pour traiter automatiquement la morphologie 2D ou 3D des pousses de plantes et en extraire des traits d’intérêt [4].

La représentation numérique du système racinaire peut représenter la topologie du l’architecture (graphe arborescent), sa géométrie et des propriétés à un instant donné ou au cours de son développement, à la fois en structure (géométrie et topologie) et en fonction (croissance, métabolisme). Étudier ce modèle et ses paramètres permet de développer des simulations de croissance racinaire, utilisées pour prédire les performances des plantes dans différentes conditions environnementales [5].

Dans la communauté scientifique, les architectures racinaires sont représentées par le format standard RootSystem Markup Language (RSML) [6], une extension du format <XML> dédiée à la description des architectures



FIGURE 2.3 – Exemple de dispositifs d’acquisition de données pour le phénotypage racinaire, utilisés dans notre étude, respectivement les cas d’étude (CE) numéro 1, 2, 7, et 4

racinaires. Ce format, qui permet des représentations informatiques des systèmes racinaires des plantes, facilitent diverses applications :

1. Partage des modèles dans un format standard **lisibles** par tous les **outils** de la communauté.
2. Analyse de la structure et de la géométrie des racines, et l’extraction de traits phénotypiques par des scripts standardisés utilisant des bibliothèques dédiées.
3. Simulation de la croissance des racines et prédiction des performances des plantes dans différentes conditions environnementales, avec la possibilité de comparer les résultats des simulations avec des modèles réels issus d’expériences de phénotypage.
4. Étude des interactions racinaires, telles que la compétition pour les ressources et la symbiose avec les micro-organismes du sol.
5. Développement de digital twin (Functional and Structural Plant Model) : répliques virtuelles détaillées des plantes permettant de simuler et d’analyser leurs comportements dans divers cadres. [7]

Méthodes de reconstruction de l'architecture racinaire existantes

Plusieurs méthodes ont été développées pour reconstruire les systèmes racinaires, et elles peuvent être classées en deux catégories principales :

- **Les méthodes manuelles**, qui nécessitent une intervention humaine pour segmenter les racines et reconstruire leur structure et leur géométrie. Par exemple, [rhizoTrak](#) [8] ou [Dart](#) [9].
- **Les méthodes automatiques**, qui utilisent des algorithmes pour segmenter les racines. Par exemple, [RootNav 2.0](#) [10] ou [RootPainter](#) [11].

Dans le cadre de l'extraction de traits phénotypiques à partir de grandes quantités de données, les méthodes automatiques sont préférées pour leur rapidité et leur reproductibilité. La communauté a initialement développé des méthodes de segmentation d'images pour détecter les racines et éliminer l'arrière-plan, suivi par des méthodes de reconstruction de l'architecture racinaire pour rechercher des chemins optimaux dans les structures segmentées [12, 13]. On a vu l'émergence de méthodes de segmentation basées sur des réseaux neuronaux convolution profonds [10-12, 14], qui montrent des progrès significatifs en matière de segmentation racinaire.

Les algorithmes de reconstruction de systèmes racinaires interprètent les ensembles de pixels segmentés comme des systèmes racinaires connectés pour produire des primitives de niveau supérieur (illustration sur la figure [b](#) ci-dessous), telles que les organes végétaux et leurs connexions [15-17] (cf. figure [c](#)), représentés informatiquement comme des *arbres* (ou forêt d'arbres) via le format RSML.

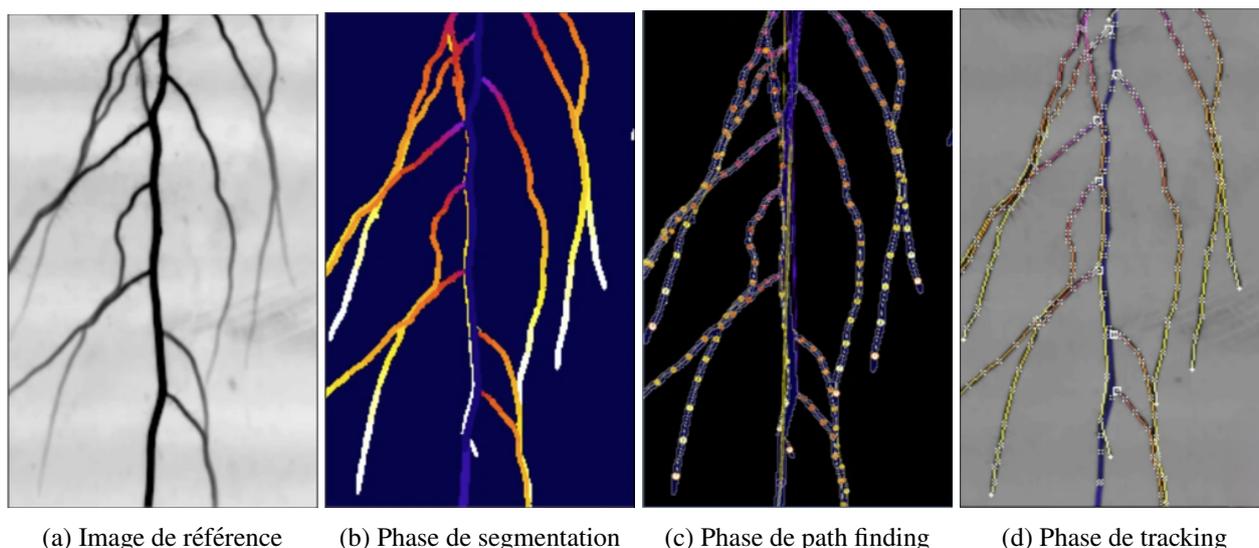


FIGURE 2.4 – Pipeline de reconstruction de l'architecture racinaire.

2.1.4 Limitations actuelles des méthodes de reconstruction de l'architecture racinaire

Plusieurs problèmes se posent quant à la reconstruction automatique de l'architecture racinaire à partir d'images :

1. Il est impossible de visualiser les structures *entières* du système racinaire **au fil du temps** dans leur environnement de croissance naturel [18].
2. Les multiples intersections et contacts entre racines, comme ceux de la figure [2.5](#), impliquent une combinatoire élevée du problème de reconstruction [10, 14, 19] car à chaque fois que deux racines se croisent, il y a deux fois plus de combinaisons possibles dans le problème de la résolution des chemins racinaires.

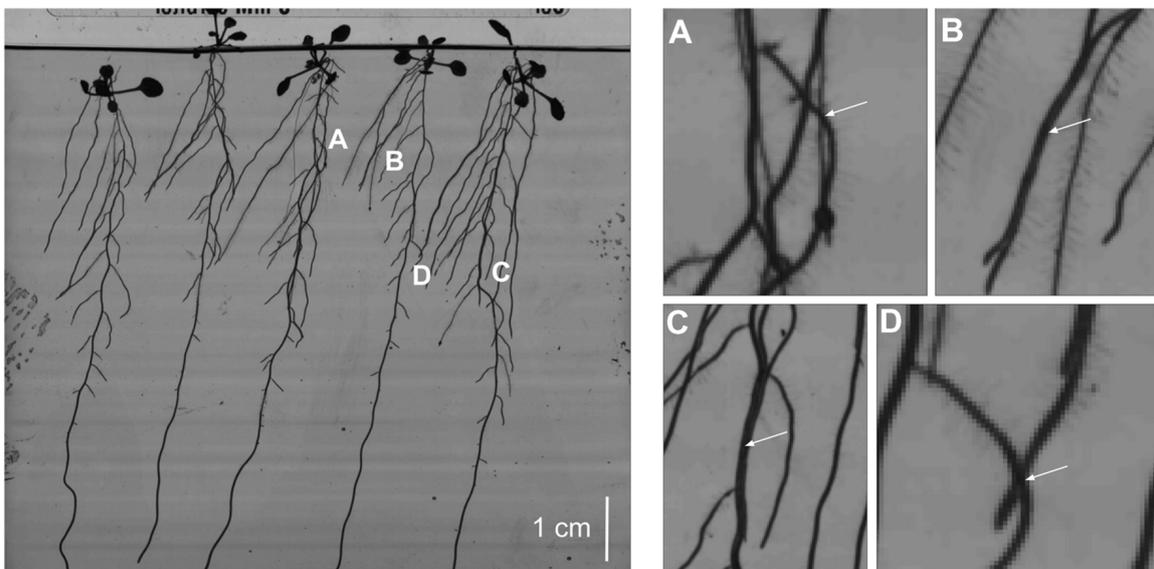


FIGURE 2.5 – Image de croisements de racines, repris de [1, (Fernandez et al., 2022)]

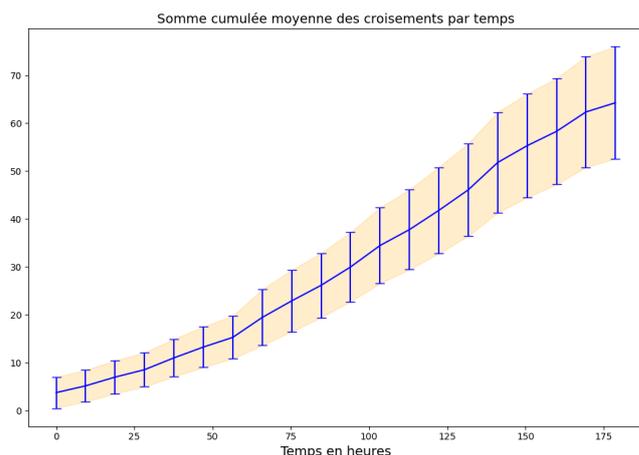


FIGURE 2.6 – Moyenne de la somme cumulée du nombre de croisements des racines au cours du temps pour 552 plantes, basée sur les données du [cas d'étude 1](#) de ce stage.

C'est d'autant plus le cas lors des expériences de suivi temporel, où le nombre de croisements entre racines augmente avec le développement de la plante (cf. figure 2.6). Les algorithmes actuels, qui reconstruisent les racines *séquentiellement*, utilisent des approches de type algorithmes glouton en faisant de la recherche de chemin optimal ([20, A*] utilisé dans RootNav 2.0[10]), ce qui conduit généralement à des optimas locaux plutôt qu'à une solution optimale globale [1].

Toutefois, [1, Fernandez et al. 2022] a montré que l'intégration de l'information temporelle est une opportunité pour résoudre ces ambiguïtés en combinant espace et temps pour distinguer les connexions organiques des croisements de racines. Les données temporelles sont déjà disponibles dans la communauté, mais l'analyse des séries temporelles reste en général sous-utilisée [18]. Le projet RootSystemTracker a été initié [1], pour ces raisons, afin de développer des méthodes de reconstruction de l'architecture racinaire basées sur le suivi topologique des racines par un algorithme d'optimisation global appliqué aux unités de croissance de racines extraites de proches en proches à *partir de séries temporelles d'images*, afin d'améliorer la fiabilité topologique des reconstructions. Cette approche a démontré son efficacité dans un cadre limité à des données capturées en conditions contrôlées, en permettant d'étudier la dynamique du développement racinaire, de garantir la topologie des architectures reconstruites, et de fournir des informations sur les taux de croissance ainsi que sur les réponses adaptatives des racines aux variations environnementales (cf. figure 2.7).

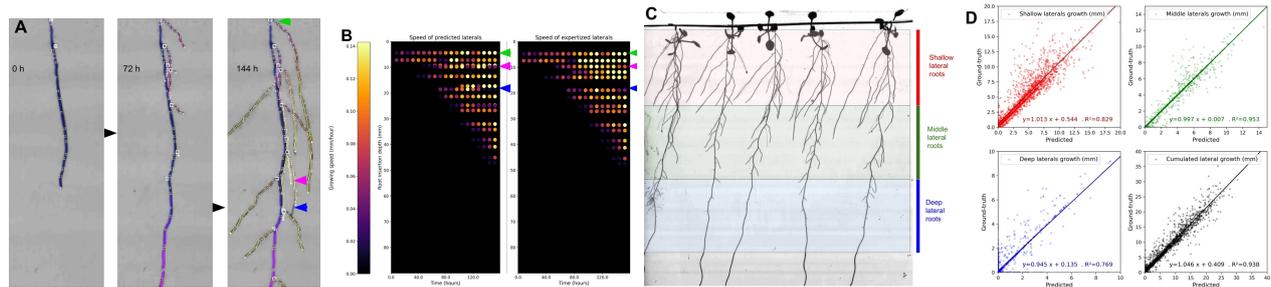


FIGURE 2.7 – Performance de la quantification des caractéristiques dynamiques de la vitesse d’émergence et d’élargissement des racines latérales en comparant les données prédites et expérimentées. Tiré de [1]

Cependant, cette méthode repose sur des hypothèses principalement vérifiées sur des données à haut débit issues de laboratoires, telles que des hypothèses restrictives sur la topologie des architectures racinaires, à savoir des racines primaires et latérales, sur la croissance des racines (plutôt linéaire et orientée vers le bas de l’image) et en l’absence de terre ou de substrat. Ces limitations restreignent son applicabilité à des conditions d’observation plus diversifiées.

2.2 Problématique générale du stage

La question centrale du projet RootSystemTracker est : **Comment concevoir un système automatique de reconstruction spatio-temporelle de l’architecture racinaire adapté à différents systèmes d’acquisition, différentes espèces et à des cas de peuplements mixtes, utilisables pour des données récoltées in situ ?**

Les travaux précédemment publiés dans mon équipe étant limités à un seul cas d’étude avec une seule plante, l’*Arabidopsis*, il faut permettre le travail avec des conditions variées, et l’adaptation de ces techniques, avec du Deep Learning par exemple. Pour travailler des modèles et les entraîner, il est nécessaire, premièrement de revisiter la manière dont sont construits les formats d’annotations actuels des données architecturales, en intégrant le cas des modèles spatio-temporels, puis deuxièmement, d’adapter les approches de reconstruction des architectures pour qu’elles soient utilisables pour des topologies quelconques et enfin, de construire une méthodologie pour la collecte et la mise à disposition des données.

Afin de nourrir ce projet, l’équipe dans laquelle je travaille a initiée la création d’un consortium international de partage de données pour le phénotypage spatio-temporel du système racinaire, visant à diversifier les sources de données qui doivent contribuer à la création d’une série de jeux de données annotés pour l’entraînement et la validation des modèles de reconstruction d’architecture racinaire, moyennant le développement d’un outil d’annotation approprié pour les arborescences racinaires en $2D + t$. Le recueil, la mise en forme des données, la méthodologie d’annotation et les premiers tests de reconstruction automatique sont les objectifs centraux de mon projet de stage.

2.3 Objectifs du Stage

Les objectifs du stage sont les suivants :

1. Créer un ensemble de jeux de données à faire annoter par les membres d’un consortium de partage de données, tous regroupés au sein d’un même benchmark.
2. Développer un outil d’annotation adapté aux données spatio-temporelles pour racines.

3. Préparer la diffusion du jeu de données en tant que data challenge pour stimuler la communauté afin de dépasser l'analyse statique des architectures et passer au phénotypage spatio-temporel.

Objectifs spécifiques

1. Faire évoluer le standard RSML pour représenter des architectures racinaires en croissance au cours du temps, incluant des organes d'ordre n .
 - (a) Développer un outil permettant la conversion entre une séquence de fichier RSML 2D (les fichiers fournis) et un fichier RSML 2D + t (le nouveau format que nous proposons).
 - (b) Développer des outils de lecture, de visualisation et d'édition des données RSML 2D + t .
2. Améliorer et déployer un outil d'annotations de séries 2D+t d'architectures racinaires. Accompagner les utilisateurs pour l'appropriation de l'outil, et développer les fonctionnalités demandées.
3. Travailler sur les données issues [des plateformes de phénotypage du consortium](#) pour les annoter et les organiser en jeux de données.
 - (a) Créer un Plan de Gestion des Données (PGD)
 - (b) Travailler sur la mise en place d'un data challenge
4. Optimiser et retravailler les algorithmes de reconstruction de l'architecture racinaire existante (en particulier la partie recalage).

Nota Bene : cas d'études

Dans la suite de ce rapport, j'évoquerai plusieurs "cas d'études" (CE). Dans le cadre de ce projet, un cas d'étude représente un ensemble cohérent de données recueillies à partir d'une ou plusieurs plateformes de phénotypage racinaire issue d'un laboratoire ou d'une collaboration. Les cas d'étude sont essentiels, car ils fournissent des exemples concrets permettant d'appliquer et de tester les méthodes développées. En utilisant des données réelles, issues de diverses conditions expérimentales, il est possible d'évaluer la robustesse et l'efficacité des algorithmes de reconstruction et d'annotations. De plus, les cas d'étude permettent de démontrer la faisabilité des approches proposées dans des contextes variés, ce qui deviendra crucial pour la généralisation des résultats à d'autres situations.

Les différents cas d'étude utilisés dans ce projet sont présentés et détaillés dans la section dédiée à la constitution du data challenge d'architectures spatio-temporelles, accompagnés d'une description des sources de données, des méthodes d'acquisition, et des défis associés à chaque cas d'étude.

Description du travail réalisé

3.1 Vue d'ensemble des actions menées et planning du stage

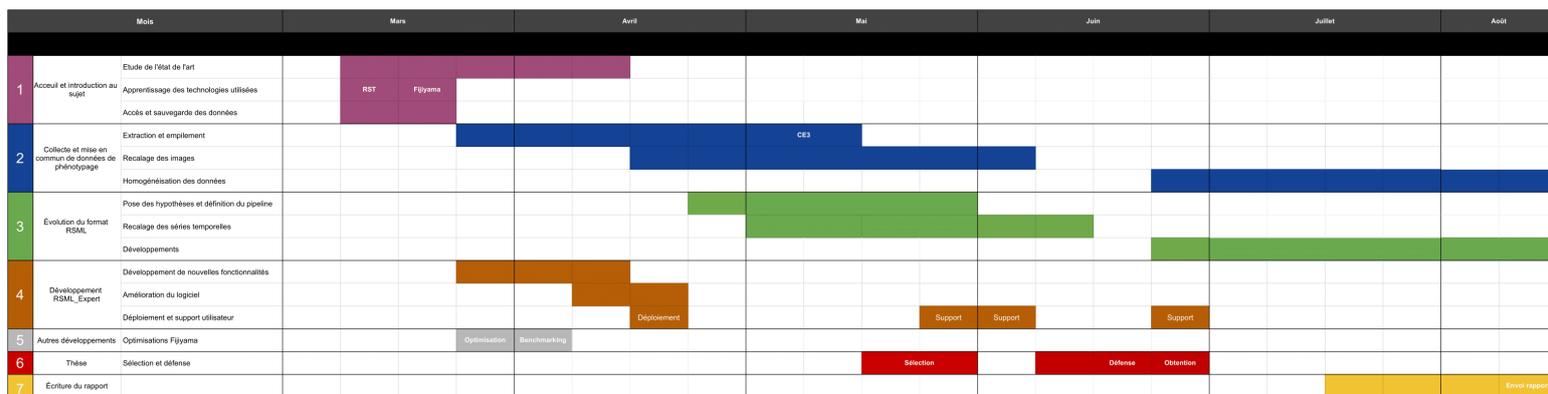


FIGURE 3.1 – Planning du stage, [lien vers le planning](#).

Mon stage, qui a duré 6 mois, a été divisé en 2 phases principales, la première phase a été consacrée à la conception de méthodes permettant le passage du format RSML 2D à un format RSML 2D + t, l'ajout d'améliorations pour la mise en place d'un outil d'annotation pour les données spatio-temporelles et l'optimisation des algorithmes de reconstruction existants, une deuxième phase consacrée à la mise en place des outils nécessaires pour la création du data challenge et la constitution des jeux de données.

J'ai également postulé à un PEPR (proposal et audition) et ai obtenu un financement de thèse dans la continuité de ce stage, pour poursuivre les travaux de recherche sur la reconstruction spatio-temporelle des architectures racinaires.

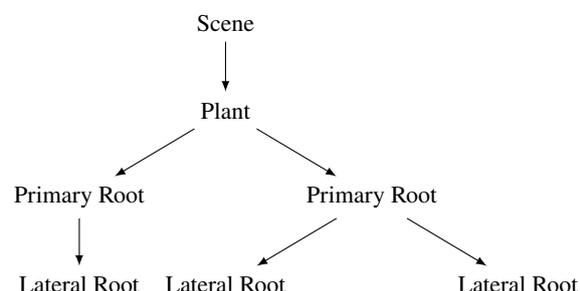
3.2 Évolution du format RSML

Le format Root System Markup Language (**RSML**) a été développé pour répondre aux besoins croissants en matière d'analyse et d'homogénéiser le formalisme de représentation des systèmes racinaires des plantes. Avec l'augmentation des outils d'analyse d'images et la diversité des formats de données qu'ils génèrent, il est devenu nécessaire de standardiser la description des architectures racinaires afin de faciliter l'interopérabilité entre différents logiciels et de permettre la comparaison et le partage de données à grande échelle.

Le format RSML, basé sur le standard **<XML>**, a été conçu pour capturer les informations géométriques, topologiques et fonctionnelles des systèmes racinaires en deux ou trois dimensions à un instant t . Il permet de décrire de manière détaillée la position spatiale des racines, les relations de parenté entre les racines primaires et latérales, ainsi que des propriétés spécifiques telles que le diamètre des racines ou la présence de poils absorbants.

3.2.1 Cadre théorique et algorithmes de conversion

La structure définie dans un fichier RSML repose sur une hiérarchie bien définie d'éléments XML qui permet de représenter les systèmes racinaires sous forme de scènes, chacune contenant une ou plusieurs plantes, et chaque plante contenant une racine primaire (racines d'ordre 1, dans les cas que j'ai pu observer) qui elle-même contient plusieurs racines latérales (racines d'ordre > 1). Les informations géométriques sont, dans les cas que j'ai étudiés, stockées sous forme de lignes brisées (polylines), tandis que les informations topologiques sont représentées par la structure imbriquée des éléments XML. Il s'agit en somme d'une arborescence définie par séquence de nœuds (ou par un format défini par l'utilisateur).



Jusqu'à maintenant, les annotations de série temporelle d'images de racines étaient représentées informatiquement par une séquence de fichiers RSML 2D (illustration sur la figure 3.2 à gauche), chacun décrivant l'architecture racinaire à l'instant où les images ont été prises. On se retrouvait donc avec plusieurs représentations numériques des architectures racinaires pour les différents temps. Ces modèles étaient parfois décalqués et/ou décalés les uns par rapport aux autres en fonction du mouvement des systèmes racinaires observés entre les acquisitions (parfois dû aux systèmes d'acquisition eux-mêmes).

L'idée de l'évolution du format vers du $2D + t$ est de permettre de représenter par *un unique modèle* une architecture racinaire en croissance au cours du temps, en ajoutant simplement une coordonnée temporelle aux nœuds de la géométrie du graphe racinaire. L'environnement fixe de la racine permet en effet de faire l'hypothèse que la racine **ne bouge pas, mais pousse par les pointes / extrémités**. Dès lors, la coordonnée temporelle associée à un nœud N peut être interprétée comme le temps de passage de la pointe racinaire à cet endroit (x, y) , donc le temps d'apparition de ce morceau de racine.

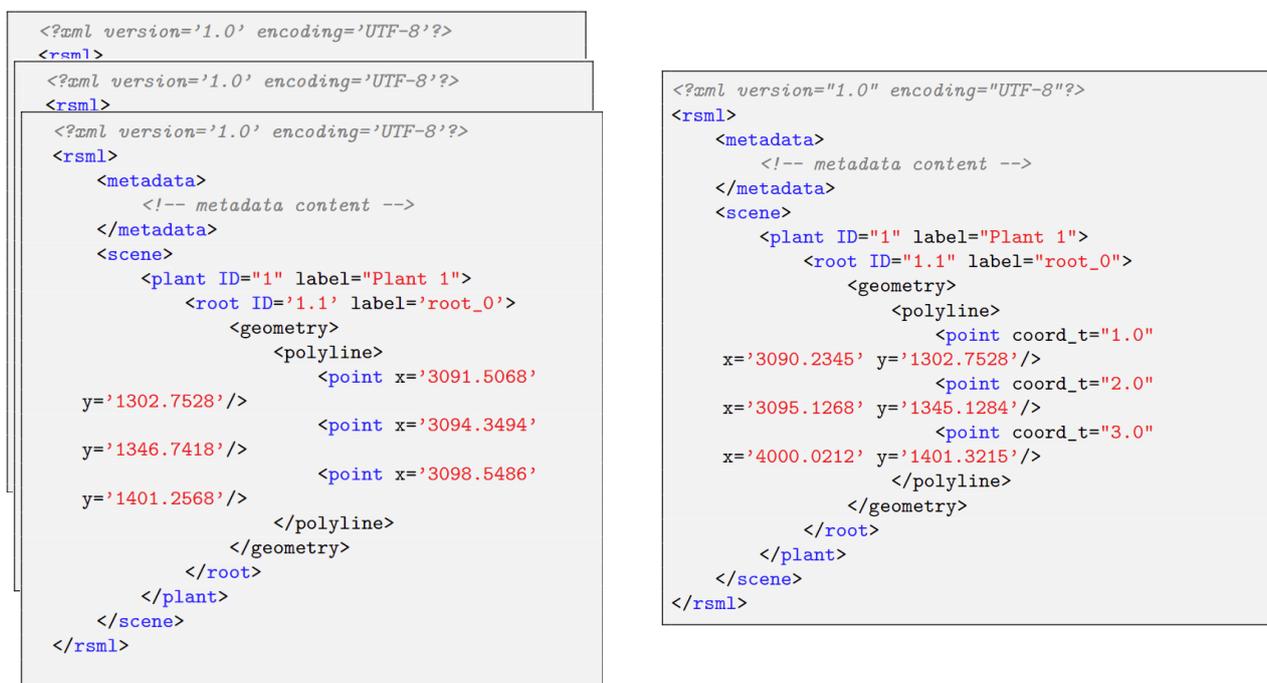


FIGURE 3.2 – À gauche une séquence de fichiers RSML 2D décrivant l'architecture racinaire à différents instants, à droite le fichier RSML $2D + t$ résultant de la conversion des fichiers 2D

Hypothèses de départ

J'ai formulé un certain nombre d'hypothèses sur les données pour pouvoir transformer une séquence temporelle de RSML $2D$ en un seul RSML $2D + t$:

1. Les annotations décrites dans les fichiers RSML sont correctes à une erreur de mesure près et décrivent les architectures racinaires observées dans les images associées au temps d'observation.
2. Toutes les racines observées continuent de croître tout au long de la période d'observation sans interruption significative, sauf en cas de mort ou d'arrêt précoce de leur croissance. On suppose qu'une racine ne "recule" pas, c'est-à-dire :

Si $l(t_1)$ est la longueur de la racine à l'instant t_1 , alors $l(t_2) \geq l(t_1)$ pour $t_2 \geq t_1$

3. Le système racinaire à un instant t , noté $S(t)$, est inclus dans le système racinaire à l'instant $t + \epsilon$. Cela signifie que pour chaque racine R_i appartenant au système racinaire $S(t)$, il existe une racine R'_i appartenant à $S(t + \epsilon)$ telle que :

(a) R_i est inclus dans R'_i . Plus formellement :

$$S(t) \subseteq S(t + \epsilon) \iff \forall R_i \in S(t), \exists R'_i \in S(t + \epsilon) \text{ tel que } G(R_i) \subseteq G(R'_i)$$

où $G(R_i) \subset \mathbb{R}^2$ est l'ensemble des coordonnées des nœuds décrivant la racine R_i .

(b) La hiérarchie au sein du système racinaire, c'est-à-dire l'ordre (primaire : 1, latérale : 2) et la parenté des racines, est conservée entre les instants t et $t + \epsilon$.

4. La vitesse de croissance de chaque racine est une fonction continue du temps $v(t)$, ce qui implique que l'on peut interpoler la position des nœuds $(x(t), y(t))$ entre deux acquisitions pour obtenir une représentation continue de la croissance racinaire.

Plus précisément, considérons deux positions successives d'une pointe d'une racine $N_{\text{préc}} = (x_{\text{préc}}, y_{\text{préc}})$ et $N_{\text{sui}} = (x_{\text{sui}}, y_{\text{sui}})$ observé respectivement aux instants $t_{\text{préc}}$ et t_{sui} , où $t_{\text{préc}} < t < t_{\text{sui}}$. La position de la racine à l'instant t peut être calculée par interpolation linéaire entre les positions précédentes et suivantes (à condition que les annotations soient suffisamment proches temporellement les unes des suivantes) :

$$x(t) = x_{\text{préc}} + \frac{(t - t_{\text{préc}})(x_{\text{sui}} - x_{\text{préc}})}{t_{\text{sui}} - t_{\text{préc}}},$$

$$y(t) = y_{\text{préc}} + \frac{(t - t_{\text{préc}})(y_{\text{sui}} - y_{\text{préc}})}{t_{\text{sui}} - t_{\text{préc}}}.$$

Cette hypothèse assure que la trajectoire des racines est une fonction continue dans le temps, sans discontinuité apparente dans l'évolution spatiale des racines.

Méthode de conversion des fichiers RSML 2D en un fichier RSML 2D + t

En me basant sur les hypothèses formulées, j'ai conçu un pipeline pour passer d'une séquence de fichiers RSML 2D à un fichier RSML $2D + t$, j'ai développé une méthode de conversion qui permet de transformer les données géométriques et topologiques décrites dans les fichiers RSML 2D vers un fichier RSML $2D + t$. L'idée générale de l'algorithme est que le modèle de la fin de série temporelle contient toutes les informations, mais on ne connaît pas le temps d'apparition de chacun des nœuds. Pour avoir cette information, il faut pouvoir suivre la croissance de chaque organe au cours du temps, et donc mettre en correspondance chaque organe "final" avec ses images dans les instants précédents. Le déroulé général est le suivant : j'utilise une méthode de recalage

d'images pour réaligner les images de la série, afin de pouvoir utiliser les transformations calculées afin de réaligner les modèles géométriques à chaque temps. Les organes étant bien superposés, je peux les mettre en correspondance au fil du temps en les comparant par une fonction de distance. Enfin, j'utilise les informations des séries temporelles de chaque organe pour déterminer le temps d'apparition de chacun des nœuds.

1. Utiliser une méthode décrite dans les données supplémentaires [21] de l'article [22, *Fujiyama, a registration tool for 3D multimodal time-lapse imaging*] pour opérer un réalignement des séries temporelles d'images, c'est à dire obtenir la meilleure superposition possible entre les images, de manière à ce que les systèmes racinaires soient comparables les uns avec les autres dans les images [suivant l'hypothèse 3](#)
2. Développer des algorithmes permettant de lire la structure hiérarchique décrite dans les annotations ainsi que les coordonnées des points qui les décrivent à chaque temps.
3. Récupérer les transformations appliquées aux images lors de leurs réalignements pour les appliquer à l'ensemble de la géométrie des systèmes racinaires. À partir de quoi on considérera que les annotations des racines satisfont les hypothèses 1 et 3.

(Développer des algorithmes permettant de trouver les racines correspondantes entre les différentes acquisitions).

4. Concevoir une méthode permettant d'interpoler les coordonnées des nœuds des racines correspondantes entre les différentes acquisitions. En satisfaisant les hypothèses 2 et 4.

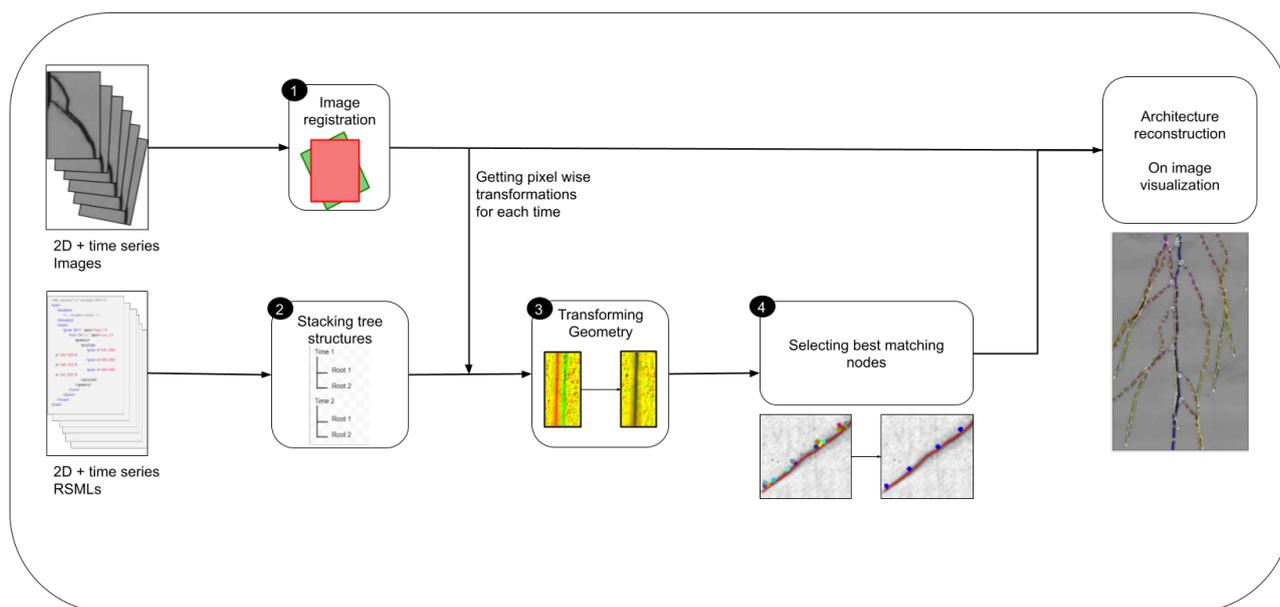


FIGURE 3.3 – Pipeline proposé pour la conversion des fichiers RSML 2D en un fichier RSML 2D + t.

Étape 1 - Recalage de la série temporelle d'images

Lors de cette première étape, on effectue un réalignement des séries temporelles d'images, en combinant plusieurs types de transformations, en particulier des translations, des rotations et des champs denses de vecteurs.

Ce réalignement des images est une étape nécessaire pour garantir que l'hypothèse 3 soit respectée. Dans le sens où l'ensemble des points qui décrivent la géométrie d'une racine à un instant t doivent également décrire la même racine à l'instant $t + \epsilon$ (Sans compter la partie qui a poussé, [hypothèse 2](#)). Ces transformations seront sauvegardées pour être plus tard appliquées à l'ensemble des nœuds des racines définies dans les fichiers RSML.

Problème de Maximisation pour le Recalage des Images

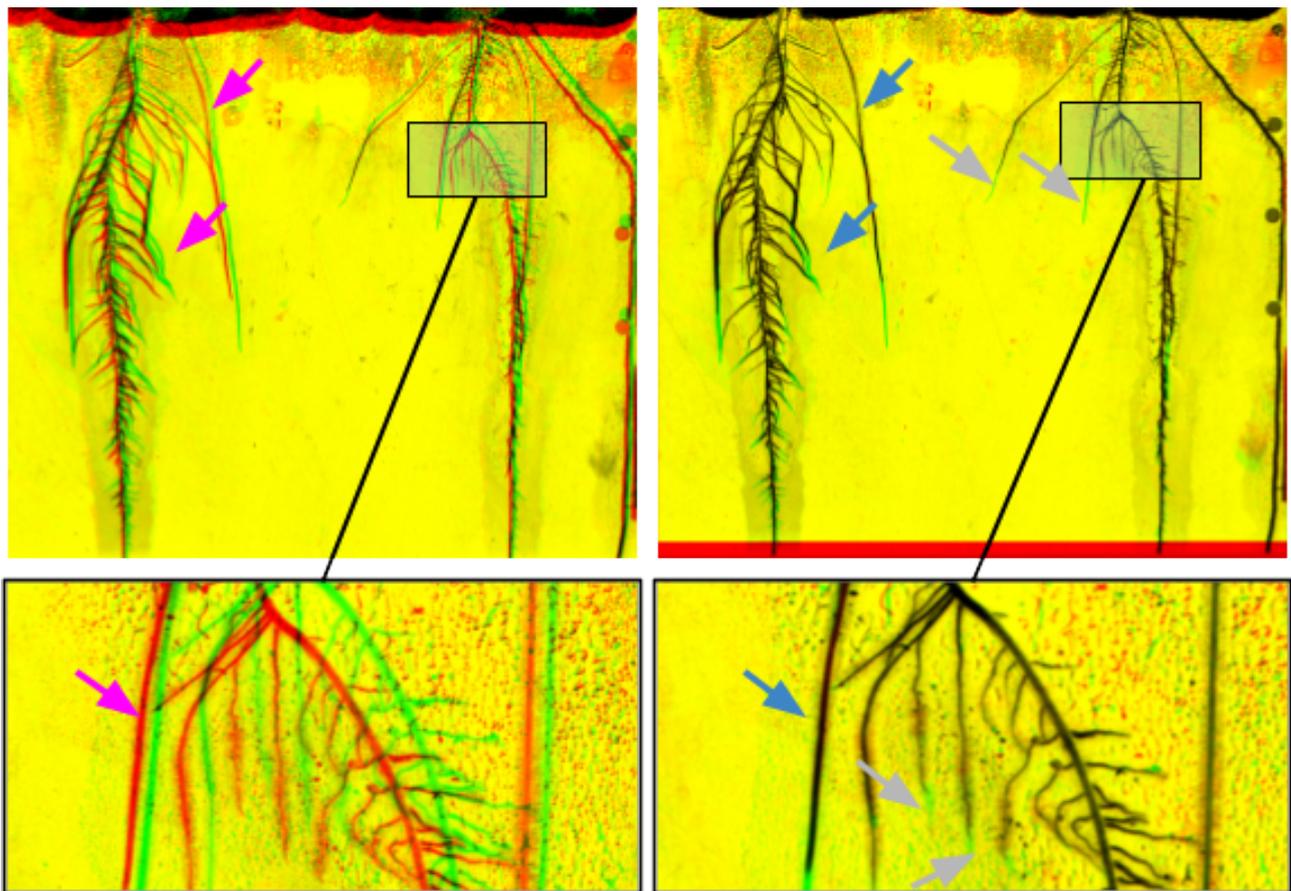


FIGURE 3.4 – Superposition d’images à recaler, extraites du cas d’étude 2. **Canal rouge** : image au temps t , **vert** : image au temps $t + 1$. À gauche : avant recalage, à droite : après recalage. L’alignement initial superpose mal les structures (**flèches magenta**), qui sont mieux superposées après recalage (**flèches bleues**). Après réalignement précis, les différences résiduelles entre les images (**flèches grises**) correspondent à la croissance des organes entre t et $t + 1$.

Comme énoncé dans [21], le recalage consiste à estimer la transformation géométrique T_{opt} qui, appliquée à une image ”mobile” I_{mov} , permet d’obtenir la meilleure superposition possible avec une image ”de référence” I_{ref} . En considérant une mesure de similarité d’image \mathcal{S} , la transformation estimée T_{opt} doit satisfaire :

$$T_{\text{opt}} = \arg \max_T \mathcal{S}(I_{\text{ref}}, I_{\text{mov}} \circ T)$$

Approche par blocs

Une approche courante pour le recalage des images est de diviser les images en blocs et de chercher des correspondances entre les blocs de l’image de référence et les blocs de l’image mobile. On définit alors n blocs B de taille $b \times b$ dans chaque image, et on cherche à maximiser un score de similarité entre les blocs correspondants.

Pour trouver les blocs correspondants entre deux images, on utilise une méthode de recherche parmi les proches voisins, qui consiste à comparer chaque bloc de l’image de référence avec les blocs de l’image mobile dans un certain voisinage autour de la position du bloc de référence. On définit une fonction de similarité \mathcal{S} , qui correspond en pratique à un coefficient de corrélation au carré, pour mesurer la similarité entre les blocs permettant in-fine de trouver le ”meilleur” bloc correspondant.

$$\forall i \in [1, n], \forall B_{\text{ref}}^i \in V(B_{\text{ref}}^i, I_{\text{ref}}), B_{\text{mov}}^i = \arg \max_{B_{\text{mov}}} \mathcal{S}(B_{\text{ref}}^i, B_{\text{mov}}) \quad (3.1)$$

Où B_{ref}^i est le bloc i présent dans l'image de référence, B_{mov} est un bloc de l'image mobile, et B_{mov}^* est le bloc mobile correspondant au bloc de référence. Les valeurs des pixels des blocs sont inscrites sous la forme de deux vecteurs ligne.

À partir des correspondances identifiées, on peut extraire un "champ sparse" de vecteurs qui peut servir à calculer une transformation d'ensemble pour l'image. Ce champ contient les vecteurs de déplacement entre les blocs de référence et les blocs correspondants dans l'image mobile.

À partir de ce champ dense de vecteurs et selon la méthode de recalage choisie :

1. **Recalage Rigide** : On estime une transformation rigide correspondante (minimisation aux moindres carrés). On définit alors une matrice de transformation rigide R telle que :

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Les images sont recalées en translation et en rotation.

2. **Recalage Dense** : On interpole le champ des déformations pour obtenir une transformation que l'on applique à l'image mobile.

Adaptation de la méthode de recalage pour les images de racines

Selon l'hypothèse 2, le système racinaire pousse entre deux acquisitions, ce qui peut affecter le recalage des images, comme montré dans la figure 3.4. Pour minimiser l'impact de cette croissance tout en respectant l'hypothèse 3 (inclusion des racines), j'ai recalé les images deux par deux (de proche en proche) par recalage rigide, afin de réduire au maximum les différences résiduelles entre les systèmes racinaires des images.

Cependant, en conditions réelles, il est possible que les racines aient légèrement bougé entre les acquisitions, nécessitant, en plus d'un recalage rigide un recalage dense pour minimiser les mouvements orthogonaux à la direction de croissance.

Le problème avec cette approche est que les racines peuvent être déformées dans le sens de leur croissance. Pour éviter ce type de déformation, j'ai dû ajuster les paramètres de l'algorithme de recalage à la main, en faisant une série de tests, pour trouver des paramètres de recalage satisfaisants. Les paramètres que j'ai eus à ajustés étaient les suivants :

- La taille des blocs
- La taille du voisinage de recherche
- La taille du pas de recherche
- Le nombre d'itérations de l'algorithme de recalage
- Le niveau de sous-échantillonnage de l'image

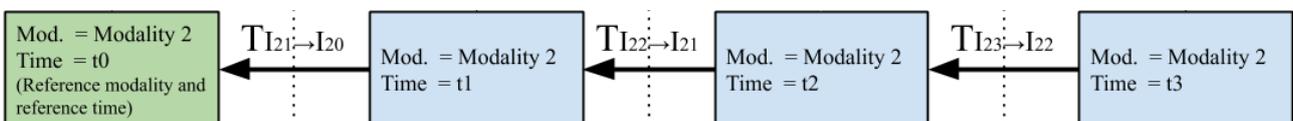


FIGURE 3.5 – Diagramme de la méthode de recalage des images 3D. Extrait de [21]

Étape 2 - Extraction des données RSML 2D

J'ai développé un algorithme permettant de lire l'ensemble des fichiers RSML 2D, les regrouper par temps, retirer les fichiers corrompus (voir les corriger quand c'était possible), puis de comparer les différentes annotations pour choisir le fichier qui satisfait au mieux les critères de validité.

Algorithm 1 RSML File Selection Algorithm

Require: *folderPath* : Path to the folder containing RSML files

Require: *dates* : Map of RSML filenames and their dates

Require: *strategy* : SelectionStrategy

Require: *removedDates* : List to store dates of removed files

Ensure: *selectedFiles* : Set of selected RSML file paths

Group files by ascending date order

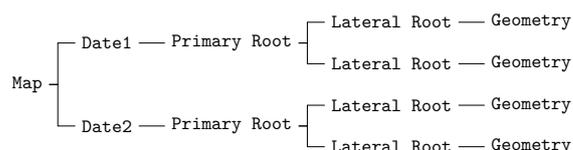
Validate and correct files if possible

Remove invalid files from groups

Apply selection strategy to choose 1 file from each group

return selected files

On obtient à la fin de cette étape une sélection de fichiers RSML 2D ordonnée en séquence, valides pour chaque temps d'observation.



Tout cela implique la création de structures de données qui associe à un temps une arborescence de racines pour un temps d'observation, un RSML_MAP.

Étape 3 - Application des transformations aux données du RSML_MAP

Les transformations calculées lors du recalage de la série d'images sont appliquées aux modèles RSML 2D correspondant pour les aligner. Cette opération va faciliter la mise en correspondance dans le temps des organes, et des nœuds qui les composent. L'application d'une transformation géométrique à un modèle RSML est une opération aisée, elle consiste à remplacer les coordonnées (x, y) de chaque nœud par les coordonnées (x', y') qu'on obtient en lui appliquant la transformation T :

$$(x', y') = T_t(x, y)$$

Les résultats obtenus après cette étape montrent que l'alignement géométrique des modèles est satisfaisant pour un premier niveau d'alignement, avec une distance moyenne résiduelle de l'ordre d'un pixel entre les nœuds correspondants. On considèrera que l'hypothèse [concernant la superposition des racines](#) est respectée, ce qui nous permet de pouvoir comparer les racines entre elles et récupérer les nœuds correspondants en considérant que les racines "sont incluses les unes dans les autres spatialement".

Boîte	mean_tot
B73_R12_01	0.83438
B73_R07_01	1.32048
B73_R05_01	1.88326
B73_R04_01	1.56113

TABLE 3.1 – Résultats sur les distances moyennes entre les nœuds de même indice aux différents temps d'observation (en pixels). Boîtes issues du [cas d'étude 3](#)

(Étape 3.5 - Associations des racines entre les différents temps d'observation)

Lors de cette étape, on essaye de mettre en correspondances les racines définies à différents temps d'observation en supposant que grâce aux étapes précédentes, on les trouvera dans une certaine proximité (un pixel, voir [tableau précédent](#)). Ce que l'on veut en sortie, c'est un ensemble de listes de racines, chaque liste contenant une racine définie à différents temps d'observation. Dans les données fournies, nous avons rencontré deux cas de figure :

1. Soit il est possible de récupérer un identifiant unique à chaque racine qui *se répète entre les* différents temps d'observation, à partir du moment où la racine est apparue.

Auquel cas l'association entre les racines des différents temps d'observation peut se faire simplement en comparant leurs identifiants.

2. Soit il y a bien un identifiant unique à chaque racine au sein d'un même fichier RSML, mais il *ne se répète pas entre* les différents temps d'observation.

Pour traiter ce second scénario, j'ai développé un algorithme permettant de comparer des racines entre elles, et de les associer en fonction de leur similarité.

Définition du Problème D'après [l'hypothèse 3](#), un système racinaire à un instant donné est inclus dans le système racinaire à l'instant suivant. L'association entre racines peut donc être formulée comme un problème d'optimisation visant à minimiser la distance (défini plus bas) entre deux racines à des instants successifs. L'objectif est de trouver une correspondance locale optimale entre les racines de deux instants successifs selon différents critères (point d'insertion, forme, correspondance des séquences temporelles), en tenant compte de la parenté et de la hiérarchie des racines.

Formellement, étant donné deux ensembles de racines à deux instants successifs, $S(t)$ et $S(t + \epsilon)$, il s'agit de déterminer une correspondance entre toutes les racines de $S(t)$ et celles de $S(t + \epsilon)$.

Méthodes de mise en correspondance entre racines Différentes méthodes peuvent être envisagées pour comparer les racines entre deux instants :

- **Méthode 1 : méthode Naïve** : comparaison des distances entre les points d'insertion des racines (le "premier" nœud). Pour chaque racine R_i de $S(t)$, on associe la racine R_j de $S(t + \epsilon)$ dont le nœud d'insertion $N_j = (x_j, y_j)$ minimise la distance euclidienne :

$$R_j^* = \operatorname{argmin}_{R_j} \|N_i - N_j\|_2.$$

- **Méthode 2 : méthode de Moyenne des Distances Cumulées** : comparaison basée sur la somme des distances entre les nœuds de même indice. Pour chaque racine R_i de $S(t)$ et R_j de $S(t + \epsilon)$, on minimise :

$$R_j^* = \operatorname{argmin}_{R_j} \frac{1}{\min(n_i, n_j)} \sum_{k=1}^{\min(n_i, n_j)} \|N_i^k - N_j^k\|_2$$

où n_i et n_j représentent le nombre de nœuds de R_i et R_j respectivement.

- **Méthode 3 : méthode combinée** : cette méthode combine les deux critères précédents avec des coefficients de pondération w_1 et w_2 :

$$R_j^* = \operatorname{argmin}_{R_j} \left[w_1 \|N_i - N_j\|_2 + w_2 \frac{1}{\min(n_i, n_j)} \sum_{k=1}^{\min(n_i, n_j)} \|N_i^k - N_j^k\|_2 \right]$$

Processus d'association des nouvelles racines : Pour toute racine R_t de $S(t)$, on cherche la racine $R_{t+\epsilon}$ de $S(t+\epsilon)$ qui maximise la similarité entre les deux racines. On stocke ensuite la racine R_t dans la liste des racines associées à $R_{t+\epsilon}$. Pour faciliter le déroulement de l'algorithme, on construit au fur et à mesure une carte de présence pour noter si cette racine a déjà été identifiée ou pas, c'est-à-dire que pour chaque racines, à chaque temps d'observation, on stocke si l'on retrouve cette racine à ce temps d'observation ou pas.

Algorithm 2 Root Correspondence Algorithm

Require: Sets of roots $\{S(t)\}_{t=1}^{N_t}$, presence map $rootPresenceMap$

Ensure: Correspondence of roots over time

```

1:  $nextTimeRoots \leftarrow S(N_t)$ 
2: for  $t \leftarrow N_t - 1$  to 1 do
3:    $currentRoots \leftarrow S(t)$ 
4:   for each  $cRoot \in currentRoots$  do
5:     if  $rootPresenceMap[cRoot.id][t + 1]$  is true then
6:       continue
7:     end if
8:      $foundRoot \leftarrow \text{findBestMatchingRoot}(cRoot, nextTimeRoots)$ 
9:     if  $foundRoot \neq \text{null}$  and  $foundRoot.label = cRoot.label$  then
10:       $cRoot.id \leftarrow foundRoot.id$ 
11:       $rootPresenceMap[foundRoot.id][t] \leftarrow \text{true}$ 
12:     end if
13:   end for
14:    $nextTimeRoots \leftarrow currentRoots$ 
15: end for
return  $\{S(t)\}_{t=1}^{N_t}$ 

```

Ici `findBestMatchingRoot` est une fonction qui prend en paramètre une racine R et une liste de racines R' , et qui renvoie la racine de R^* qui maximise la similarité avec R suivant l'une des méthodes définies précédemment. Et on répète cette opération pour chaque racine de $S(t)$, pour chaque temps t .

Les résultats obtenus pour les différentes méthodes de correspondance des racines pour une des boîtes du [cas d'étude 3](#) sont présentés dans la table suivante :

Boîte	Méthode	Pourcentage de racines avec mauvaise ou sans correspondance	Total de racines
Boîte 4	Comparaison des points d'insertion	3,7 %	1109
	Moyenne des distances cumulées	3,8 %	1109
	Méthode combinée : $w_1 = 0.1, w_2 = 0.9$	1,08 %	1109
Boîte 7	Comparaison des points d'insertion	13 %	1127
	Moyenne des distances cumulées	4,79 %	1127
	Méthode combinée : $w_1 = 0.1, w_2 = 0.9$	4,61 %	1127
Boîte 12	Comparaison des points d'insertion	1,5 %	1194
	Moyenne des distances cumulées	1,2 %	1194
	Méthode combinée : $w_1 = 0.1, w_2 = 0.9$	0,8 %	1194

TABLE 3.2 – Résultats de la mise en correspondance des racines des boîtes 4, 7, et 12 (tout temps confondus). Le plus bas est le mieux.

$$\text{resultat} = \frac{\text{nombre total de racines mal classifiées}}{\text{nombre total de racines à associer}} \times 100$$

On obtient, au mieux, un pourcentage d'erreur de correspondance entre les racines de l'ordre de 1% pour les différentes méthodes de correspondance.

Les résultats obtenus montrent que la méthode combinée avec des coefficients de pondération $w_1 = 0.1$ et $w_2 = 0.9$ (trouvées empiriquement) combinées à des conditions de validité sur les racines semblent être la plus efficace pour associer les racines entre les différents temps d'observation, avec un taux de racines avec une erreur de correspondance compris entre 0,8% et 4,61%. Sois en pratique 10 et 52 racines mal associées pour des boîtes de respectivement 1194 et 1127 racines.

Un examen plus [approfondi des résultats](#) montre que les racines mal associées sont généralement des racines qui viennent d'apparaître, ou qui sont très proches spatialement les unes des autres, ce qui rend leur association difficile.

Une première piste d'amélioration serait de réaliser un "grid search" pour trouver des coefficients de pondération optimaux pour la méthode combinée. Néanmoins, de part la façon dont le problème a été posé, à savoir une résolution d'un problème de minimisation pour des correspondances entre racines de proche en proche, une erreur d'association pour un instant t peut se propager à l'ensemble des racines correspondantes précédentes. De plus, on recherche une correspondance locale optimale entre les racines de deux instants successifs, sans garantie de correspondance globale. L'utilisation d'une méthode de résolution globale du problème d'association des racines pourrait être une piste d'amélioration pour garantir une correspondance globale optimale entre les racines.

Étape 4 - Interpolation des nœuds entre les différents temps d'observation

L'étape précédente fournit une liste de racines associées entre les différents temps d'observation. Il faut le convertir en une seule racine comprenant tous les points géométriques correspondant à la progression de la pointe racinaire, avec la coordonnée temporelle correspondant au point de passage de la pointe racinaire.

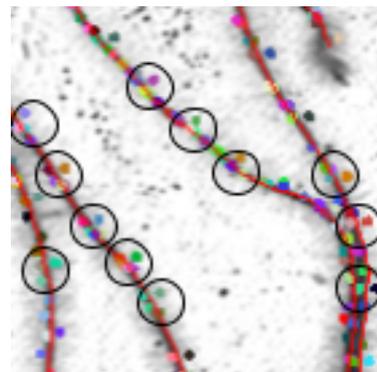


FIGURE 3.6 – Chaque point coloré sur l'image est un nœud d'une racine. Les cercles regroupent les nœuds d'une même racine par temps d'apparition. Image extraite du cas [d'étude 3](#).

L'idée est de créer une racine avec des nœuds informés en temps tel que $R = \{N_1, N_2, \dots, N_n\}$, où $N_i = (x_i, y_i, t_i)$, t_i étant le temps d'observation associé au nœud N_i . C'est fait en suivant les étapes suivantes :

1. **Sélection de la racine la plus longue** : En temps normal, c'est la racine annotée au dernier temps d'observation, parfois l'avant dernier temps (en cas d'erreur d'annotation).
2. **Identification du temps d'apparition de chacun des nœuds** : on associe la longueur progressive de la racine avec le temps. Puis on mesure la distance entre la base de la racine et chaque nœud (dans toute la série temporelle). On peut alors déterminer le temps d'apparition de chacun des nœuds par interpolation linéaire, en particulier le temps d'apparition de chacun des nœuds de la racine sélectionnée à l'étape 1.

Longueur à partir de laquelle le nœud est considéré comme apparu.

3. **Ré-interpolation de la racine sélectionnée** : on a associé un temps à chaque nœud (la valeur du temps peut être un nombre à virgule). Pour garantir qu'on fournit la position exacte de la pointe racinaire à chaque temps d'observation, on construit les nœuds manquants en calculant leur position (x, y) par interpolation

linéaire.

3.2.2 Outil logiciel pour les annotations de RSML temporels : RSML_Expert

Mon travail de développement logiciel a consisté à ajouter et à améliorer les fonctionnalités de RSML_Expert et à le rendre plus maintenable. Ce travail s'inscrit dans un contexte de génie logiciel visant à développer un outil utilisable notamment par des chercheurs en biologie végétale.

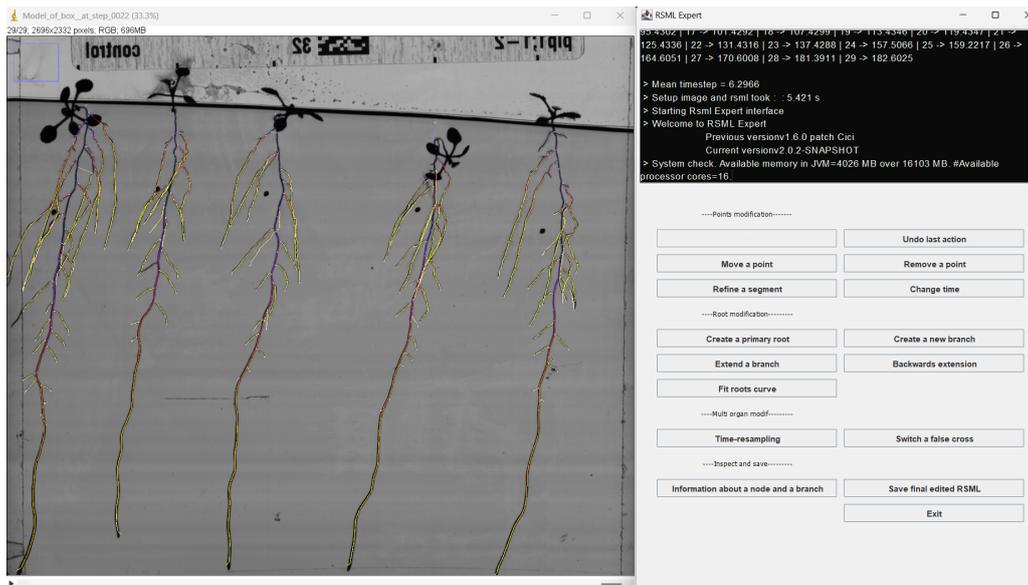


FIGURE 3.7 – Interface graphique de l'outil RSML_Expert pour l'annotation des racines.

J'ai utilisé des méthodes de développement itératives basées sur les pratiques de programmation agile orientées vers le test (Test Driven Development, TDD), en collaboration avec l'équipe de RootSystemTracker. J'ai été amené à travailler sur des Forks du projet, à développer des fonctionnalités, à les tester, puis à les intégrer dans le projet principal après validation de l'équipe.

L'outil d'annotation était initialement intégré au logiciel de recalage d'images [Fijiyama](#) (projet indépendant), j'ai engagé un refactoring pour extraire l'outil d'annotation du packaging de Fijiyama, en modifiant les appels de fonctions spécifiques à Fijiyama et j'ai migré le code vers un nouveau dépôt [RootSystemTracker](#). Cette migration visait à transformer RSML_Expert visait à clarifier la gestion des projets.

Travail sur la fonctionnalité principale de RSML_Expert : annoter

Les annotations des racines, dans la série temporelle d'images, se font en demandant à l'utilisateur d'annoter une série de points $P = p_1, p_2, \dots, p_n$ tels que $p_i = (x_i, y_i, t_i)$, où t_i est le temps d'acquisition associée à l'image en cours d'annotation. Nous avons posé certaines conditions de validité sur les points annotés pour garantir la cohérence des données :

1. **Ordre chronologique** : $t_i \leq t_{i+1}$ pour tout $i \in [1, n - 1]$.
2. **Au moins un point par temps d'acquisition** : $t_{i+1} - t_i \leq 1$ pour tout $i \in [1, n - 1]$, assurant qu'au moins un point est associé à chaque image de la série à partir du moment où la racine devient visible.

Les racines sont ensuite créées en reliant successivement les points, avec un ajustement temporel s'il y a plusieurs points pour un même temps. En pratique, le dernier point ajouté d'une série correspond à la pointe de la racine

pour le temps d'acquisition donné, tandis que les autres points sont considérés comme des points intermédiaires [suivant l'hypothèse 4](#). Ces opérations se réalisent sur une interface graphique simplifiée développée avec les bibliothèques de ImageJ, afin de s'appuyer sur ses capacités pour l'imagerie $2D + t$ (gestion de stacks d'image) et son appropriation large par la communauté des biologistes.

Fonctionnalités ajoutées

Il existe déjà des fonctionnalités pour ajouter, supprimer et déplacer des points, ainsi que pour créer des racines latérales. J'ai ajouté ces fonctionnalités dans RSML_Expert :

- **Création de racines primaires** : permet de définir des racines primaires à partir d'une série de points ordonnés chronologiquement.
- **Extension des racines en sens inverse** : ajout de segments à des racines existantes en suivant l'ordre inverse chronologique (*backward*). Cette fonctionnalité permet de compléter les racines (primaires ou latérales) dont les points de départ / d'insertion n'ont pas été annotés correctement.

Les difficultés rencontrées lors de l'ajout de ces fonctionnalités étaient liées à la gestion des points extrêmes des racines et leur association avec les racines déjà existants, la cohérence entre le code que j'ai pu rajouter et l'ensemble du projet (TDD).

Autres travaux de développement

En plus des fonctionnalités précédemment mentionnées, j'ai également travaillé sur divers aspects du logiciel :

- L'optimisation des performances en réduisant le temps de chargement des images et des annotations grâce à la parallélisation des tâches indépendantes.
 - J'ai utilisé du *multithreading* pour paralléliser certaines opérations lourdes telles que la lecture de plusieurs fichiers RSML ou la copie des pixels d'images.
 - J'ai employé des classes tels que `ExecutorService` ou des implémentations de l'interface `Stream` pour répartir les tâches sur plusieurs threads en fonction du nombre de cœurs disponibles sur la machine.
- La gestion des fichiers RSML $2D + t$ pour la lecture et l'écriture des annotations, en tirant parti du format XML sous-jacent.

En optant plutôt pour une lecture par balise XML plutôt que ligne par ligne via des expressions régulières, méthode employée dans la version initiale.

- J'ai utilisé la bibliothèque `javax.xml.parsers` qui est une bibliothèque standard et maintenue. Étant native en Java, elle garantit une compatibilité à long terme et ne pose à priori aucun problème de dépendance externe.
- La mise en place de classes de tests unitaires pour assurer la fiabilité des nouvelles fonctionnalités.
 - J'ai utilisé `JUnit5` pour écrire des séries de tests unitaires.

Les tests sont configurés pour vérifier chaque fonctionnalité : création de racines, modification de points, etc.

3.3 Préparations d'un data challenge d'architecture spatio-temporelle

Afin d'évaluer de manière objective les méthodes de reconstruction spatio-temporelle des systèmes racinaires, mon équipe a proposé la création d'un data challenge. Ce challenge a pour but de comparer différentes approches

de reconstruction des systèmes racinaires à partir d'images $2D + t$ et de données annotées en RSML $2D + t$.

Dans ce cadre, j'ai contribué à la constitution du jeu de données en assemblant des données issues de plusieurs cas d'étude (CE), en mettant en place un plan de gestion des données.

3.3.1 Description du challenge

Le challenge consiste à pouvoir évaluer et à comparer différentes méthodes de reconstruction spatio-temporelle des systèmes racinaires à partir d'images capturées par diverses plateformes de phénotypage. Les participants devront reconstruire les systèmes racinaires à partir des images et des annotations fournies. Les principaux défis incluent :

- La gestion des grandes quantités de données
- L'hétérogénéité des formats et des résolutions d'images
- La variabilité des structures racinaires étudiées en conditions naturelles/ou non.

3.3.2 Constitution du jeu de donnée

L'une des premières étapes cruciales de ce projet a consisté à exploiter les données issues d'un consortium formé avant le début du stage. Les données proviennent des différentes plateformes de phénotypage mentionnées précédemment. Par la suite, nous avons discuté avec les membres du consortium pour sélectionner des cas d'étude et caractériser les données.

Labo/UC	Libellé jeux de donnée	Débit	Méthode d'acquisition	Espèces	Ordre racinaires	# Plantes	# Boîtes	# Times-teps	Ambassadeur	Institut	UMR
CE1-BPMP	CE1-BPMP	HTP	AgarBox	Arabidopsis	2	5	32	29	Amandine Crabos	INRAE	BPMP
CE2-JOURDAN	CE2-JOURDAN_IS	LTP	In-situ	Arachide, Mil, Sorgho	4	-	222	47-162	Christophe Jourdan	CIRAD	Eco&Sols
	CE2-JOURDAN_RZT	MTP	Rhizotron	Arachide, Mil, Sorgho	4	-	876	108-130			
CE3-LEPSE	CE3-LEPSE_RHIZOTRON	MTP	BuvarBox	Maïs	2	2	4	13-14	Bertrand Muller	INRAE	LEPSE
CE4-RHIZOSCOPE	CE4-RHIZOSCOPE	MTP	Rhizoscope	Riz	4	1	2	400	Thomas Laisne	CIRAD	
CE5-VALENTIN	CE5-VALENTIN	MTP	Rhizotube	Riz	5	3	3	21 - 22	Gaetan Louarn	INRAE (Lusignan)	UR P3F
CE6-ECOSYS	CE6-ECOSYS	MTP	Rhizotron	Blé	2	-	2	8-9	Tristan Gérault	INRAE	Ecosys
CE7-CIAT	CE7-CIAT	MTP	Rhizotube	-	5	-	18	9	Camila Reboledo	CIAT	CIAT

TABLE 3.3 – Tableau des cas d'études du consortium.

Description de l'organisation du jeu de données après réorganisation

J'ai regroupé les données d'images dans plusieurs répertoires, chacun correspondant à un cas d'étude spécifique. Chaque répertoire contient désormais des dossiers organisés comme suit :

- Les données brutes sont stockées dans le dossier RAW, telles qu'elles nous ont été transmises ou suite à une réorganisation et un renommage.
- L'ensemble des données traitées et annotées sont/seront stockées dans le dossier FULL_DATA.
- Un sous-ensemble représentatif de FULL_DATA est stocké dans le dossier KICK_DATA à destination du data challenge.
- Les scripts utilisés pour le (pré)traitement des données sont conservés dans le dossier SCRIPTS.

Toute correction ou modification apportée à la main a été enregistrée dans un fichier markdown associé à chaque jeu de données, permettant de suivre l'historique des modifications et de garantir la reproductibilité des résultats. De même, l'ordre dans lequel exécuter les scripts est indiqué dans un fichier markdown associé au sein du même dossier.

3.3.3 Description des étapes de préparation des données

La préparation du jeu de données peut être résumé en plusieurs phases :

1. Organisation des fichiers
2. Création de séries temporelles d'images
3. Réalignement des séries temporelles d'images en suivant les étapes décrites dans la section [Recalage sur les images](#).
4. Correction des erreurs à la main.
5. Annotation des données par les [membres du consortium](#).

Extraction et empilement des données

CE1 déjà organisé

Dans le cas du CE1, aucune préparation des données n'était requise, le pipeline était déjà existant depuis 2022 [1]. Ce cas d'étude m'a servi de référence pour la mise en place des autres cas d'études.

Dans le cas général (CE2 et CE4 à CE7) :

La préparation des données dans le cas général a été réalisée en plusieurs étapes :

1. État des lieux des fichiers en présence et de la structure des données.
2. Récoltes des métadonnées associées à chaque image, soit à l'issue d'un script python permettant de les extraire (Commande `exiftool`), soit directement depuis un fichier CSV.
3. Association des images avec les temps d'acquisition, pour chaque boîtes et pour chaque plante.

Gestion des données pour l'CE3 :

Pour le CE3, le travail mené a été un peu plus complexe. Les images étaient initialement organisées en trois dossiers distincts, chacun correspondant à une méthode d'acquisition différente (deux dossiers pour des minirhizotrons et un pour les scanners). Elles étaient uniquement différenciées par leur nom de fichier, lequel inclut des informations telles que le numéro du puits dans lequel évolue la plante, un numéro associé à la plante et la date d'acquisition de l'image. Le principal problème de ces données étant le manque d'homogénéité générale. J'ai identifié plusieurs problèmes :

— Le volume important de données à traiter :

Minirhizotron 1 : ~ 73 plantes x 124 temps.

Minirhizotron 2 : ~ 60 plantes x 11 temps.

Scanner : ~ 37 plantes x 164 temps.

— Des erreurs potentielles dans les noms de fichiers (corrigées à la main).

— Des métadonnées manquantes.

— Une variabilité importante dans la convention de nommage des fichiers (~ 110 motifs différents), sachant qu'il s'agissait du seul moyen qui m'a été indiqué pour différencier les images.

- Des résolutions et formats d'image pouvant varier au sein d'une même série temporelle (scanner).

J'ai d'abord développé plusieurs scripts Python pour réorganiser chacun des jeux de données de l'CE3 en une arborescence de fichiers plus structurée (C'est cette arborescence que j'ai décidé de charger dans le dossier RAW du CE3). Les scripts incluent la création d'une liste des nomenclatures des fichiers, des renommages ainsi que du pattern matching sur les noms de fichiers pour extraire les informations nécessaires.

L'arborescence finale des fichiers était organisée de la manière suivante : PUIITS → SCAN → DATE → IMAGE TEMPORELLE.

Ensuite, j'ai conçu un script pour empiler en série temporelle les images pour chaque date d'un même puits et d'une même plante. Il m'a fallu traiter deux problèmes spécifiques :

- En pratique, il pouvait y avoir plusieurs images pour une même date
- Les tailles des images pouvaient varier, certaines longueurs ou largeurs d'images pouvaient avoir ~ 2–5% de pixels en plus ou en moins les unes par rapport aux autres (problématique pour l'empilement).

Il était donc nécessaire de s'assurer de l'homogénéité des tailles des images avant empilement, puis de sélectionner parmi la sélection d'images celles qui étaient les plus similaires aux images des dates voisines (autrement dit celles qui s'insèrent le mieux localement dans la série temporelle).

À ce moment-là du stage, je n'étais pas totalement familier avec l'outil de recalage des images et ai donc opté pour des choix radicaux qui à posteriori ne semblent pas optimaux, mais qui permettent malgré tout de construire des séries temporelles d'images un tant soit peu cohérentes :

1. Lister l'ensemble des images d'une même plante, récupérer les tailles d'images les plus fréquentes et redimensionner l'ensemble images en conséquence.

Un des problèmes de cette approche est la variation possible des tailles des racines entre deux images même si contenu et très faible.

2. Utiliser un algorithme de comparaison d'images (coefficients de corrélation identique à celui utilisé pour Fijiyama, mais appliqué à l'ensemble des pixels des deux images) pour sélectionner les images les plus similaires.

Enfin, un post-traitement manuel des données a été réalisé pour résoudre les derniers problèmes spécifiques.

3.3.4 Plan de gestion des données - FAIR

Le plan de gestion des données que j'ai mis en place tente d'adhérer aux principes FAIR (Findable, Accessible, Interoperable, and Reusable) pour garantir que les données et les outils développés sont largement utilisables par la communauté scientifique. Voici comment chaque principe est appliqué dans le cadre de ce projet :

Dans le cadre du projet RootSystemTracker, un plan de gestion des données (PGD) a été mis en place pour garantir que les données produites et utilisées dans le projet soient facilement accessibles, réutilisables et conformes aux bonnes pratiques FAIR (Findable, Accessible, Interoperable, and Reusable). Le respect de ces principes est essentiel pour assurer une gestion rigoureuse des données tout au long du projet et faciliter leur partage avec la communauté scientifique.

Findable (Facile à trouver)

- **Section unique dédiée à chaque cas d'étude** : Chaque banque de données est organisée dans un répertoire dédié contenant les données brutes, les données traitées et les scripts utilisés pour le traitement des données.

- **Métadonnées descriptives** : Pour l'ensemble des données, j'ai collecté des métadonnées telles que les informations sur les espèces, les méthodes d'acquisition, les conditions de croissance, les intervalles de temps, etc. Décrites dans un fichier [CSV](#).
- **Indexation et référencement** : Les données pour l'instant sont stockées sur le serveur [Meso-lr](#) de l'Université de Montpellier sur le cluster MUSE.

Les données, une fois annotées, seront indexées dans des bases de données reconnues et des archives ouvertes telles que [Zenodo](#).

Accessible (Facile d'accès)

- **Accessibilité des données** : Les données ont été stockées sur un serveur de stockage dédié et accessible via un lien de téléchargement.

Interopérable (Interopérable)

- **Utilisation de formats ouverts** : Les métadonnées fournies ont été stockées dans des formats ouverts tels que CSV ou RSML et les images temporelles ont été stockées dans le format d'image TIFF.
- **Vocabulaire et standards communs** :
 - Chaque image des séries d'images temporelles comprend un titre au format :
`<date_d'acquisition>t<temps_d'acquisition` résultant de l'association temporelle des images.
 - Chaque banque de série d'images est accompagnée de markdown décrivant les étapes de préparation des données.
 - Chaque banque de série d'images est accompagnée de markdown décrivant les modifications apportées à la main aux données.
 - Chaque banque de série d'images est organisée en répertoires RAW, KICK_DATA et FULL_DATA.
 - Certaines des banques de données sont accompagnées d'un dossier SCRIPTS contenant l'ensemble des scripts utilisés pour le traitement des données.
- **Interopérabilité des logiciels** : Les outils développés, tels que...
 - Ceux permettant la conversion de séquence d'images en séries temporelles d'images.
 - Ceux permettant la conversion de fichiers RSML (format déjà utilisé par la communauté) vers RSML $2D + t$ont été développés en Java et sont compatibles avec les systèmes d'exploitation Windows et Linux.

Reusable (Réutilisable)

- **Documentation** : Historique des modifications faites à la main dans un fichier markdown, ajout de scripts utilisant des librairies standard et maintenues (Java, Python).

3.4 Optimisations et améliorations des outils de recalage d'images

Une des problématiques liées au réaligement des images était la lenteur de l'outil FijiYama lors des phases d'application des transformations géométriques. Plus précisément, le système existant nécessitait, pour appliquer une transformation géométrique à une image, de charger en mémoire un objet Java de type ImageITK. Or, la librairie utilisée jusqu'à présent dans le projet était ImageJ. Les méthodes de conversion d'un objet ImageJ en ImageITK consistaient :

- Soit en une conversion pixel par pixel de l'image (ce qui était très coûteux en temps de calcul).

- Soit en une sauvegarde de l'image sur le disque dur / ssd, suivie d'un rechargement de celle-ci sous un autre format (ce qui peut être coûteux en temps d'écriture et de lecture).

Le problème ne provenait pas tant du développement que de la librairie utilisée, à savoir `SimpleITK`, prise via Maven. Il s'agissait d'une version non officielle de `SimpleITK`, non maintenue et non mise à jour, qui n'offrait pas la possibilité d'accéder aux buffers des images (toujours utilisé).

J'ai donc proposé une solution alternative consistant à utiliser une version plus récente de la librairie (même si cela nécessite de revoir les modes de déploiement des mises à jour des logiciels existants), permettant l'utilisation d'un buffer pour stocker les images en mémoire et les manipuler directement depuis ce buffer. J'ai donc travaillé sur l'implémentation de méthodes statiques permettant de convertir des images `ImageJ` en `SimpleITK` et vice-versa, en utilisant des buffers de différents types (byte, short, char, float, double).

Nombre de temps	Largeur	Hauteur	Temps d'exécution (ms)
$N_t = 1$			
1	512	512	1362,9 ± 143,9
1	1024	1024	1455,2 ± 311,3
1	2048	2048	3226,2 ± 146,8
$N_t = 5$			
5	512	512	1617,7 ± 296,7
5	1024	1024	3075,9 ± 223,2
5	2048	2048	8941,4 ± 709,2
$N_t = 10$			
10	512	512	1984,6 ± 241,7
10	1024	1024	5363,8 ± 637,8
10	2048	2048	15893,5 ± 1637,8
$N_t = 20$			
20	512	512	2838,0 ± 389,7
20	1024	1024	10359,2 ± 2479,2
20	2048	2048	27563,8 ± 2031,8
$N_t = 50$			
50	512	512	5513,8 ± 629,2
50	1024	1024	20428,2 ± 3827,9
50	2048	2048	72976,0 ± 6578,9
$N_t = 100$			
100	512	512	9443,1 ± 490,5
100	1024	1024	35475,0 ± 2900,8
$N_t = 200$			
200	512	512	17470,0 ± 2650,8
200	1024	1024	69056,0 ± 6668,1

TABLE 3.4 – Temps d'exécutions des nouvelles méthodes de transfert par buffer pour des résolutions et nombres d'itérations différents. Moyenne pour 100 images en 32 bits

Les résultats (cf. 3.5) montrent une nette amélioration des temps d'exécution grâce à l'utilisation de buffers pour le stockage en mémoire des images. Les anciennes méthodes, reposant sur la sauvegarde sur disque ou la conversion pixel par pixel, montrent des limitations importantes, notamment pour les images si le nombre de temps est élevé.

Pour des images de taille 512×512 pixels, avec 5 temps, on passe de 158, 172 s pour 200 temps avec la méthode de sauvegarde sur disque à 17, 470 s pour la même configuration, soit une méthode 9, 05 fois plus rapide (24, 6 fois plus rapide que la méthode par copie pixel par pixel).

Ce travail impacte directement les performances de l'outil de recalage d'images, utilisé pour la reconstruction spatio-temporelle des architectures racinaires de `RootSystemTracker`, notamment utilisé par le cas d'étude 1.

Nombre de temps	Largeur	Hauteur	Temps d'exécution (ms)
Méthode par sauvegarde sur disque			
1	512	512	1016,0 ± 71,38
5	512	512	3376,2 ± 409,40
10	512	512	5623,6 ± 430,76
20	512	512	11528,0 ± 1172,76
50	512	512	29389,0 ± 1607,22
100	512	512	69705,8 ± 7075,60
200	512	512	158172,4 ± 9492,81
Méthode de copie, pixel par pixel			
1	512	512	2007,0 ± 98,86
5	512	512	9992,6 ± 240,52
10	512	512	20228,8 ± 337,82
20	512	512	42347,6 ± 3160,68
50	512	512	108105,4 ± 11374,82
100	512	512	211333,6 ± 20238,09
200	512	512	430344,0 ± 49665,68

TABLE 3.5 – Temps d'exécutions des anciennes méthodes de transfert pour des résolutions et nombres d'itérations différents. Moyenne pour 100 images en 32 bits

Conclusion et perspectives

Au terme de ce stage de Master 2, mon travail, en tant que stagiaire ingénieur de recherche, m'a permis de contribuer de manière significative au projet RootSystemTracker, notamment en optimisant les méthodes de reconstruction spatio-temporelle des architectures racinaires à partir de données de série temporelle d'images. En outre, j'ai développé de nouveaux algorithmes de conversion des fichiers RSML vers un format $2D+t$, mis en place un outil logiciel pour l'annotation de ces données, et constitué un jeu de données robuste pour le lancement d'un data challenge international. Les résultats obtenus montrent que mes apports ont permis d'améliorer la vitesse et la portée des reconstructions racinaires, ouvrant ainsi la voie à des analyses plus approfondies de la dynamique racinaire en conditions réelles.

La formation reçue à Polytech Nice Sophia a été essentielle pour mener à bien ce projet. Les compétences acquises en mathématiques appliquées, en science des données et en modélisation m'ont permis d'aborder les défis rencontrés.

L'impact de ce travail donne lieu à une prolongation en thèse, dans le cadre de laquelle je m'attacherai à renforcer les modèles développés en intégrant des techniques de deep learning et en exploitant les données issues du data challenge pour améliorer et étendre l'application encore la reconstruction spatio-temporelle des systèmes racinaires. Cette thèse offrira l'opportunité de transférer ces avancées du laboratoire au champ, favorisant ainsi une meilleure compréhension des interactions racinaires dans des contextes agroécologiques variés.

En conclusion, ce stage a été une étape déterminante dans mon parcours académique, me permettant de contribuer à la préparation d'un projet de recherche de doctorat. Le travail réalisé ouvre des perspectives prometteuses pour l'amélioration des techniques de phénotypage racinaire, avec des applications directes pour l'agriculture durable et la gestion des ressources naturelles, de même, il a constitué une première étape de ma carrière professionnelle.

Poursuite en thèse

J'ai postulé (proposal et audition devant jury) et ai obtenu un financement pour le projet "RootSystemTracker" dans le cadre du Programme Prioritaire de Recherche (PEPR) Agroécologie et Numérique (26 proposal reçus, 6 sélectionnés). Ce projet vise à développer des méthodes automatiques de reconstruction spatio-temporelle de l'architecture racinaire des plantes en utilisant l'intelligence artificielle (IA) pour améliorer la conception de systèmes agroécologiques.

Résumé du projet de thèse : Les racines des plantes jouent un rôle essentiel dans l'absorption des nutriments et de l'eau, la fixation du carbone atmosphérique et l'interaction avec les sols. Cependant, les méthodes actuelles de phénotypage racinaire sont limitées par des problèmes d'occlusion et de variabilité des conditions d'observation. Le projet "RootSystemTracker" surmonte ces limitations en utilisant des algorithmes de graphes et des techniques de deep learning pour analyser les trajectoires racinaires.

L'objectif principal est de transférer ces techniques du laboratoire au champ, en utilisant des données produites par un consortium international et partagées sous forme de data challenges. Ce projet renforcera la coopération internationale et encouragera l'accessibilité des ressources, permettant aux pays du Nord et du Sud de bénéficier des avancées méthodologiques. Les résultats attendus incluent le développement de logiciels open-source pour l'annotation spatio-temporelle et la reconstruction racinaire, ainsi que la publication de plusieurs articles scientifiques.

Résumé

Ce rapport rend compte du travail réalisé lors de mon stage de Master 2 au sein de l'équipe PhenoMen de l'UMR AGAP au CIRAD à Montpellier, consacré à la reconstruction spatio-temporelle des architectures racinaires à partir de séries temporelles d'images. J'ai œuvré principalement à proposer une évolution du format RSML vers une version $2D + t$, permettant de décrire par un modèle numérique la croissance de l'architecture racinaire dans le temps, et j'ai ainsi développé des algorithmes pour la conversion entre l'ancien format (2D) et le nouveau. En outre, j'ai poursuivi le développement de l'outil logiciel RSML.Expert, en enrichissant et en ajoutant des fonctionnalités d'annotations. J'ai également constitué et mis en forme un jeu de données en vue de publier un data challenge, intégrant des informations issues de diverses plateformes de phénotypage. Le rapport détaille également la création d'un plan de gestion des données et les optimisations logicielles significatives au traitement de ces données lors de la reconstruction des architectures racinaires. Ces travaux visent à transposer les méthodes développées du laboratoire au champ, avec des implications directes pour la recherche agronomique et agro-écologique. Ce stage fut pour moi une opportunité de mettre au défi mes compétences académiques et m'a permis de m'engager dans un projet de recherche novateur. Ce travail va se poursuivre en thèse.

Abstract

This report describes the work carried out during my Master's 2 internship in the PhenoMen team of the UMR AGAP at CIRAD in Montpellier, devoted to the spatio-temporal reconstruction of root architectures from time series of images. I primarily worked on proposing an evolution of the RSML format towards a $2D + t$ version, making it possible to describe the growth of root architecture over time using a digital model, and I also developed algorithms for converting between the old format (2D) and the new one. Moreover, I pursued the development of the software tool RSML.Expert, by improving and adding annotation functionalities. I also compiled and formatted a dataset with a view to publishing a data challenge, integrating information from various phenotyping platforms. The report also details the creation of a data management plan and the significant software optimisations for processing this data when reconstructing root architectures. The aim of this work is to transfer the methods developed in the laboratory to the field, with direct implications for agronomic and agro-ecological research. For me, this internship was an opportunity to challenge my academic skills and to get involved in an innovative research project. This work will be followed up by a thesis.

Bibliographie

1. FERNANDEZ, R., CRABOS, A., MAILLARD, M., NACRY, P. & PRADAL, C. High-throughput and automatic structural and developmental root phenotyping on Arabidopsis seedlings. en. *Plant Methods* **18**, 127. ISSN : 1746-4811. <https://plantmethods.biomedcentral.com/articles/10.1186/s13007-022-00960-5> (2024) (déc. 2022).
2. JOHNSON, M., TINGEY, D., PHILLIPS, D. & STORM, M. Advancing fine root research with minirhizotrons. en. *Environmental and Experimental Botany* **45**, 263-289. ISSN : 00988472. <https://linkinghub.elsevier.com/retrieve/pii/S0098847201000776> (2024) (juin 2001).
3. NGUYEN, H. A. *et al.* Are high-throughput root phenotyping platforms suitable for informing root system architecture models with genotype-specific parameters ? An evaluation based on the root model ArchiSimple and a small panel of wheat cultivars. en. *Journal of Experimental Botany* **75** (éd. DUPUY, L.) 2510-2526. ISSN : 0022-0957, 1460-2431. <https://academic.oup.com/jxb/article/75/8/2510/7634095> (2024) (avr. 2024).
4. DAVIET, B., FERNANDEZ, R., CABRERA-BOSQUET, L., PRADAL, C. & FOURNIER, C. PhenoTrack3D : an automatic high-throughput phenotyping pipeline to track maize organs over time. en. *Plant Methods* **18**, 130. ISSN : 1746-4811. <https://plantmethods.biomedcentral.com/articles/10.1186/s13007-022-00961-4> (2024) (déc. 2022).
5. LANDL, M. *et al.* Measuring root system traits of wheat in 2D images to parameterize 3D root architecture models. en. *Plant Soil* **425**, 457-477. ISSN : 0032-079X, 1573-5036. <http://link.springer.com/10.1007/s11104-018-3595-8> (2024) (avr. 2018).
6. LOBET, G. *et al.* Root System Markup Language : Toward a Unified Root Architecture Description Language. en. *Plant Physiol.* **167**, 617-627. ISSN : 0032-0889, 1532-2548. <https://academic.oup.com/plphys/article/167/3/617-627/6113575> (2024) (mars 2015).
7. MITSANIS, C., HURST, W. & TEKINERDOGAN, B. A 3D functional plant modelling framework for agricultural digital twins. en. *Computers and Electronics in Agriculture* **218**, 108733. ISSN : 01681699. <https://linkinghub.elsevier.com/retrieve/pii/S0168169924001248> (2024) (mars 2024).
8. MÖLLER, B. *et al.* rhizoTrak : a flexible open source Fiji plugin for user-friendly manual annotation of time-series images from minirhizotrons. en. *Plant Soil* **444**, 519-534. ISSN : 0032-079X, 1573-5036. <http://link.springer.com/10.1007/s11104-019-04199-3> (2024) (nov. 2019).
9. LE BOT, J. *et al.* DART : a software to analyse root system architecture and development from captured images. en. *Plant Soil* **326**, 261-273. ISSN : 0032-079X, 1573-5036. <http://link.springer.com/10.1007/s11104-009-0005-2> (2024) (jan. 2010).
10. YASRAB, R. *et al.* RootNav 2.0 : Deep learning for automatic navigation of complex plant root architectures. en. *GigaScience* **8**, giz123. ISSN : 2047-217X. <https://academic.oup.com/gigascience/article/doi/10.1093/gigascience/giz123/5614712> (2024) (nov. 2019).
11. SMITH, A. G. *et al.* Root Painter : deep learning segmentation of biological images with corrective annotation. en. *New Phytologist* **236**, 774-791. ISSN : 0028-646X, 1469-8137. <https://nph.onlinelibrary.wiley.com/doi/10.1111/nph.18387> (2024) (oct. 2022).
12. MOLLER, B., SCHRECK, B. & POSCH, S. *Analysis of Arabidopsis Root Images — Studies on CNNs and Skeleton-Based Root Topology* en. in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)* (IEEE, Montreal, BC, Canada, oct. 2021), 1294-1302. ISBN : 978-1-66540-191-3. <https://ieeexplore.ieee.org/document/9607677/> (2024).

13. NAGEL, K. A. *et al.* The platform GrowScreen-Agar enables identification of phenotypic diversity in root and shoot growth traits of agar grown plants. en. *Plant Methods* **16**, 89. ISSN : 1746-4811. <https://plantmethods.biomedcentral.com/articles/10.1186/s13007-020-00631-3> (2024) (déc. 2020).
14. GAGGION, N. *et al.* ChronoRoot : High-throughput phenotyping by deep segmentation networks reveals novel temporal parameters of plant root system architecture en. Oct. 2020. <http://biorxiv.org/lookup/doi/10.1101/2020.10.27.350553> (2024).
15. RELLÁN-ÁLVAREZ, R., LOBET, G. & DINNENY, J. R. Environmental Control of Root System Biology. en. *Annu. Rev. Plant Biol.* **67**, 619-642. ISSN : 1543-5008, 1545-2123. <https://www.annualreviews.org/doi/10.1146/annurev-arplant-043015-111848> (2024) (avr. 2016).
16. RUFFEL, S. *et al.* Nitrogen economics of root foraging : Transitive closure of the nitrate–cytokinin relay and distinct systemic signaling for N supply vs. demand. en. *Proc. Natl. Acad. Sci. U.S.A.* **108**, 18524-18529. ISSN : 0027-8424, 1091-6490. <https://pnas.org/doi/full/10.1073/pnas.1108684108> (2024) (nov. 2011).
17. MAUREL, C. & NACRY, P. Root architecture and hydraulics converge for acclimation to changing water availability. en. *Nat. Plants* **6**, 744-749. ISSN : 2055-0278. <https://www.nature.com/articles/s41477-020-0684-5> (2024) (juin 2020).
18. WEIHS, B. J. *et al.* The State of the Art in Root System Architecture Image Analysis Using Artificial Intelligence : A Review. en. *Plant Phenomics* **6**, 0178. ISSN : 2643-6515. <https://spj.science.org/doi/10.34133/plantphenomics.0178> (2024) (jan. 2024).
19. SYMONOVA, O., TOPP, C. N. & EDELSBRUNNER, H. DynamicRoots : A Software Platform for the Reconstruction and Analysis of Growing Plant Roots. en. *PLoS ONE* **10** (éd. DE SMET, I.) e0127657. ISSN : 1932-6203. <https://dx.plos.org/10.1371/journal.pone.0127657> (2024) (juin 2015).
20. DORAN, J. E., MICHIE, D. & KENDALL, D. G. Experiments with the Graph Traverser program. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences* **294**, 235-259. eprint : <https://royalsocietypublishing.org/doi/pdf/10.1098/rspa.1966.0205>. <https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1966.0205> (1966).
21. FERNANDEZ, R. & MOISY, C. Supplementary data for : Fijiyama : a registration tool for 3D multimodal time-lapse imaging. en. *Bioinformatics* **37** (éd. JINBO, X.) 1482-1484. ISSN : 1367-4803, 1367-4811. <https://academic.oup.com/bioinformatics/article/37/10/1482/5913388> (2024) (juin 2021).
22. FERNANDEZ, R. & MOISY, C. Fijiyama : a registration tool for 3D multimodal time-lapse imaging. en. *Bioinformatics* **37** (éd. JINBO, X.) 1482-1484. ISSN : 1367-4803, 1367-4811. <https://academic.oup.com/bioinformatics/article/37/10/1482/5913388> (2024) (juin 2021).

Annexes : lien vers l'appel à projet et le proposal

Lien vers l'appel à projet : <https://aap-pepr.agorize.com/fr/challenges/agroekonum-aapdocpostdoc2024>

Lien vers le proposal : https://docs.google.com/document/d/1G3CMeTrnBVsvwAHcCHk0owXO_tTABNVbD_vqxnirso/edit?usp=sharing

Annexe : Résultats

Git repository : [RootSystemTracker](#), branch Dev_Loai

Date	Méthode	Racines sans correspondances	Racines mal classifiées	Total de racines
2018-05-13	Comparaison point insertion	1	0	17
	Distance moyenne	2	0	17
	Combinaison (pt insertion + distance)	2	0	17
2018-05-14	Comparaison point insertion	3	0	31
	Distance moyenne	2	0	31
	Combinaison (pt insertion + distance)	2	0	31
2018-05-15	Comparaison point insertion	4	0	40
	Distance moyenne	2	0	40
	Combinaison (pt insertion + distance)	2	0	40
2018-05-16	Comparaison point insertion	7	0	60
	Distance moyenne	3	0	60
	Combinaison (pt insertion + distance)	2	0	60
2018-05-17	Comparaison point insertion	12	0	81
	Distance moyenne	5	0	81
	Combinaison (pt insertion + distance)	4	0	81
2018-05-18	Comparaison point insertion	13	0	90
	Distance moyenne	4	0	90
	Combinaison (pt insertion + distance)	4	0	90
2018-05-19	Comparaison point insertion	14	0	109
	Distance moyenne	4	0	109
	Combinaison (pt insertion + distance)	4	0	109
2018-05-20	Comparaison point insertion	14	0	120
	Distance moyenne	4	0	120
	Combinaison (pt insertion + distance)	4	0	120
2018-05-21	Comparaison point insertion	15	0	132
	Distance moyenne	6	0	132
	Combinaison (pt insertion + distance)	6	0	132
2018-05-22	Comparaison point insertion	15	0	139
	Distance moyenne	4	0	139
	Combinaison (pt insertion + distance)	4	0	139
2018-05-23	Comparaison point insertion	15	0	144
	Distance moyenne	4	0	144
	Combinaison (pt insertion + distance)	4	0	144
2018-05-24	Comparaison point insertion	7	0	146
	Distance moyenne	3	0	146
	Combinaison (pt insertion + distance)	3	0	146
2018-05-25	Comparaison point insertion	0	0	146
	Distance moyenne	0	0	146
	Combinaison (pt insertion + distance)	0	0	146

TABLE 4.1 – Comparaison des résultats de classification des racines par méthode et par date. Boîte 4.

Date	Méthode	Racines sans correspondances	Racines mal classifiées	Total de racines
2018-05-13	Point d'insertion	1	0	17
	Distance moyenne	2	0	17
	Combinaison (pt insertion + distance)	2	0	17
2018-05-14	Point d'insertion	3	0	31
	Distance moyenne	2	0	31
	Combinaison (pt insertion + distance)	2	0	31
2018-05-15	Point d'insertion	4	0	40
	Distance moyenne	2	0	40
	Combinaison (pt insertion + distance)	2	0	40
2018-05-16	Point d'insertion	7	0	60
	Distance moyenne	3	0	60
	Combinaison (pt insertion + distance)	2	0	60
2018-05-17	Point d'insertion	12	0	81
	Distance moyenne	5	0	81
	Combinaison (pt insertion + distance)	4	0	81
2018-05-18	Point d'insertion	13	0	90
	Distance moyenne	4	0	90
	Combinaison (pt insertion + distance)	4	0	90
2018-05-19	Point d'insertion	14	0	109
	Distance moyenne	4	0	109
	Combinaison (pt insertion + distance)	4	0	109
2018-05-20	Point d'insertion	14	0	120
	Distance moyenne	4	0	120
	Combinaison (pt insertion + distance)	4	0	120
2018-05-21	Point d'insertion	15	0	132
	Distance moyenne	6	0	132
	Combinaison (pt insertion + distance)	6	0	132
2018-05-22	Point d'insertion	15	0	139
	Distance moyenne	4	0	139
	Combinaison (pt insertion + distance)	4	0	139
2018-05-23	Point d'insertion	15	0	144
	Distance moyenne	4	0	144
	Combinaison (pt insertion + distance)	4	0	144
2018-05-24	Point d'insertion	7	0	146
	Distance moyenne	3	0	146
	Combinaison (pt insertion + distance)	3	0	146
2018-05-25	Point d'insertion	0	0	146
	Distance moyenne	0	0	146
	Combinaison (pt insertion + distance)	0	0	146

TABLE 4.2 – Comparaison des résultats de classification des racines par méthode et par date. Boîte 7.

Date	Méthode	Racines sans correspondances	Racines mal classifiées	Total de racines
2018-05-12	Comparaison point insertion	3	0	4
	Distance moyenne	3	0	4
	Combinaison (pt insertion + distance)	3	0	4
2018-05-13	Comparaison point insertion	3	0	28
	Distance moyenne	2	0	28
	Combinaison (pt insertion + distance)	1	0	28
2018-05-14	Comparaison point insertion	2	0	44
	Distance moyenne	2	0	44
	Combinaison (pt insertion + distance)	1	0	44
2018-05-15	Comparaison point insertion	4	0	65
	Distance moyenne	3	0	65
	Combinaison (pt insertion + distance)	2	0	65
2018-05-16	Comparaison point insertion	4	0	74
	Distance moyenne	3	0	74
	Combinaison (pt insertion + distance)	2	0	74
2018-05-17	Comparaison point insertion	1	0	94
	Distance moyenne	0	0	94
	Combinaison (pt insertion + distance)	0	0	94
2018-05-18	Comparaison point insertion	2	0	108
	Distance moyenne	1	0	108
	Combinaison (pt insertion + distance)	1	0	108
2018-05-19	Comparaison point insertion	2	0	116
	Distance moyenne	1	0	116
	Combinaison (pt insertion + distance)	1	0	116
2018-05-20	Comparaison point insertion	0	0	123
	Distance moyenne	0	0	123
	Combinaison (pt insertion + distance)	0	0	123
2018-05-21	Comparaison point insertion	0	0	130
	Distance moyenne	0	0	130
	Combinaison (pt insertion + distance)	0	0	130
2018-05-22	Comparaison point insertion	0	0	136
	Distance moyenne	0	0	136
	Combinaison (pt insertion + distance)	0	0	136
2018-05-23	Comparaison point insertion	0	0	136
	Distance moyenne	0	0	136
	Combinaison (pt insertion + distance)	0	0	136
2018-05-24	Comparaison point insertion	0	0	136
	Distance moyenne	0	0	136
	Combinaison (pt insertion + distance)	0	0	136

TABLE 4.3 – Comparaison des résultats de classification des racines par méthode et par date. Boîte 12.

Annexe : Technologies Utilisées

Langages

- **Python** : Utilisé principalement pour le traitement des données, la création de scripts d'automatisation, et l'analyse des images. Voir le site officiel : python.org.
- **Java** : Développement de l'outil d'annotation `RSML_Expert` et du pipeline de conversion `RSML 2D` vers `RSML 2D + t`. Voir le site officiel : java.com.

Bibliothèques et Outils

- **Fijiyama** : Librairie pour le recalage d'images multi-modales et l'analyse spatio-temporelle. Plus d'informations : github.com/Rocsg/Fijiyama.
- **SimpleITK** : Utilisée pour les transformations d'images, notamment dans le recalage d'images. Documentation officielle : simpleitk.org.
- **ImageJ/Fiji** : Outil de traitement d'images avec une interface graphique, utilisé pour l'analyse des images et la visualisation des résultats. Voir : imagej.net.
- **Meso-Ir Muse** : Plateforme utilisée pour l'hébergement et le traitement de gros volumes de données d'images. Plus d'informations : meso-lr.umontpellier.fr.

Outils Collaboratifs et Standards

- **SSH** : Utilisé pour accéder aux serveurs de calcul distants, transférer des fichiers, et exécuter des commandes à distance. Voir : ssh.com.
- **Git** : Système de gestion de versions pour le suivi des modifications de code et la collaboration. Voir : git-scm.com.
- **RSML** : Format standard de description des architectures racinaires, utilisé pour les échanges et l'annotation des données. Plus d'informations : rootssystemml.github.io.

Autres Technologies Utilisées

- **Maven** : Gestionnaire de projets et de dépendances pour Java, utilisé pour compiler et gérer les bibliothèques dans `RSML_Expert`. Voir : maven.apache.org.
- **JUnit** : Bibliothèque de tests unitaires en Java, utilisée pour assurer la qualité du code dans `RSML_Expert`. Voir : junit.org/junit5.

Laboratoire

CIRAD (Centre de coopération internationale en recherche agronomique pour le développement)

- └ CIRAD Montpellier
 - └ Département BIOS
 - └ UMR AGAP
 - └ Équipe Phenomen

Mentions légales

Français :

- Ce travail a bénéficié d'une aide de l'État gérée par l'Agence Nationale de la Recherche au titre de France 2030 portant la référence ANR-16-CONV-0004.
- Ce travail a été réalisé avec le soutien de la plateforme MESO@LR de l'Université de Montpellier.

English :

- This work was supported by the French National Research Agency under France 2030 program, referred as ANR-16-CONV-0004.
- This work has been realized with the support of MESO@LR-Platform at the University of Montpellier.