

Reconstitution d'inventaires forestiers avec le scanner laser terrestre LiDAR.



ANNEE : 2010

N° du PE : CL-2009-169

CENTRE DE RATTACHEMENT PE : Cluny.

AUTEUR : Etienne TRICOT.

TITRE : Reconstitution d'inventaires forestiers avec le scanner laser terrestre LiDAR (Light Detection And Ranging).

ENCADREMENT DU PE : Alexandre PIBOULE, Office National des Forêts, département Recherche
Michael KREBS, ARTS, LaBoMaP.

Rémy MARCHAL, Arts et Métiers ParisTech, LaBoMaP.

PARTENAIRE DU PE : ONF (Office National des Forêts), département Recherche, Velaine en Haye.

NOMBRE DE PAGES : 85 (dont 50 pour le rapport).

NOMBRE DE REFERENCES BIBLIOGRAPHIQUES : 5.

RESUME : Le PE proposé par l'ONF porte sur la valorisation du scanner laser terrestre LiDAR en foresterie.

Compte-tenu des applications « classiques » de l'outil, il s'agit d'évaluer la capacité de l'outil à reconstruire un inventaire forestier.

Le projet comprend ainsi la réalisation d'un logiciel de reconstruction d'inventaires par l'intermédiaire d'algorithmes qu'il faudra assembler, évaluer, corriger voire créer, l'étude du gain réel apporté par le scanner laser terrestre, l'étude des autres possibilités apportées par l'outil (réalisation de Modèles Numériques de Terrain locaux, mesure de hauteurs, d'essences...).

MOTS CLES : FORESTERIE/LIDAR/INVENTAIRE/FORET/RECONSTRUCTION/SCANNER LASER
TERRESTRE.

PARTIE A REMPLIR PAR LE PROFESSEUR RESPONSABLE DU PROJET

ACCESSIBILITE DE CE RAPPORT (entourer la mention choisie)

Classe 0 = accès libre.

Classe 1 = Confidentiel jusqu'au __/__/____.

Classe 2 = Hautement confidentiel.

Date :

Nom du signataire :

Signature :

Remerciements

Je tiens, en premier lieu, à remercier mon tuteur de stage, Alexandre Piboule, non seulement pour sa disponibilité, mais surtout pour son investissement dans ce projet, et en particulier en ce qui concernait les aspects de codage.

Je salue par ailleurs l'ensemble de l'équipe du site R&D de Velaine-en-Haye que je souhaite remercier pour sa bonne humeur et surtout pour son sens de l'accueil :

Jérôme Bock pour son sérieux indéfectible.

Jean-Pierre Renaud pour son pragmatisme à toute épreuve.

Régis Bindner et Nicolas Gomez pour leur maîtrise de R.

Michel Magoux pour ses dons d'ubiquité.

Olivier Lanter pour ses conseils lors du choix des parcelles de test.

Bénédicte Rochet pour sa bonne humeur permanente.

Je remercie bien évidemment l'ensemble des intervenants du laboratoire Bois de Cluny :

Rémy Marchal, professeur encadrant du projet, pour son écoute, malgré l'éloignement de ce stage avec les problématiques du LaBoMap.

Laurent Bléron, pour ses idées lors de l'élaboration des algorithmes de reconstruction.

Enfin, je termine par Michael Krebs, sans qui ce projet n'aurait jamais pu évoluer, et qui n'a pas hésité à s'investir plus que nécessaire malgré sa non implication initiale au projet. Merci à toi !

Avis au lecteur

Ce rapport contient certains termes techniques spécifiques à la foresterie, à la sylviculture ou l'informatique, qui sont nécessaires à sa bonne compréhension.

Ces termes seront ainsi inscrits dans le corps du texte en italique et expliqués en annexe 0.

Sommaire

Avis au lecteur	5
I. Introduction.....	9
1) Le projet EMERGE.....	9
2) L’Office National des Forêts	10
3) Problématique du stage.	10
II. Etat de l’art.....	11
1) Inventaires forestiers.....	11
a) Objectif	11
b) Données obtenues.	11
c) Méthodes de mesure.	12
d) Limites actuelles.	12
2) Le LiDAR (Light Detection And Ranging).....	13
a) Présentation de l’outil.....	13
b) Utilisation dans l’industrie et la recherche.	14
c) Le scanner laser terrestre.....	15
d) Application dans le domaine forestier.	16
3) Le traitement des données LiDAR.	17
III. Matériel et méthodes.....	17
A) Matériel et participants au projet.....	17
1) Caractéristique du TLS (Terrestrial Laser Scanner).....	17
2) Outils informatiques utilisés.....	18
3) Les partenaires externes au projet.....	19
4) Mise en œuvre d’un jeu de validation en forêt de Bezange.	20
B) Développement du logiciel COMPUTREE.....	22
1) Premières analyses des jeux de données.....	22
2) Algorithmes mis en œuvre.	23
a) Séparation du nuage en tranches.	23
b) Détermination des formes.	24
c) Reconstruction des cercles et des ellipses.	25
d) Reconstruction des arbres.....	26
e) Visualisation des résultats.....	26
3) Evaluation des performances.....	27

a)	Travail sur une placette permanente.....	27
b)	Développement sous C++ et travail avec des arbres témoin.....	33
c)	Evaluation sur des jeux de données complets.....	37
IV.	Résultats et discussions.....	41
1)	Version « finale » de COMPUTREE.....	41
2)	Résultats de la campagne de validation.....	43
V.	Conclusion et perspectives.....	52
1)	Résultats.....	52
2)	Si c'était à refaire.....	53
3)	Perspectives.....	53
	Bibliographie.....	55
	Annexe 0 : termes techniques.....	56
	Annexe I : Exemple de réseau de placettes permanentes.....	57
	Annexe II : méthode d'inventaire relascopique.....	58
	Annexe III : Carte du réseau de placettes de la forêt de Bezange.....	59
	Annexe IV : Protocole d'expérimentation en forêt de Bezange.....	60
I.	Objectif de l'expérience.....	60
II.	Caractéristiques de l'expérience.....	60
III.	Localisation de l'expérience.....	60
IV.	Matériel nécessaire.....	60
V.	Protocole.....	61
VI.	Description des étapes.....	61
VII.	Planning de l'expérience.....	62
	Annexe V : interface du logiciel COMPUTREE développé en JAVA.....	63
	Annexe VI : Algorithme de calcul des diamètres des arbres reconstruits sur la version JAVA.....	64
	Annexe VII : évaluation des paramètres de COMPUTREE.....	66
	Annexe VIII : algorithmes de comparaison des données développés sous R.....	67
	Annexe IX : version finale de COMPUTREE.....	69
	Annexe X : Algorithme de localisation des arbres de la forêt de Bezange, développés sous R.....	74
	Annexe XI : cartes de la campagne de mesure en forêt de Bezange.....	78

I. Introduction.

1) Le projet EMERGE.

La consommation énergétique est, depuis plusieurs années, l'une des préoccupations principales des états. Il ne s'agit pas uniquement de consommer moins, mais surtout de consommer mieux et de produire de l'énergie différemment. C'est ainsi que de nombreuses mesures ont été prises en ce sens, en particulier via le Grenelle de l'Environnement ou encore les accords de Kyoto : elles concernent, entre autres, les transports, le bâtiment ou l'industrie.

Il en va de même pour les ressources énergétiques dites renouvelables, qu'il faut désormais d'autant plus valoriser. En particulier, les ressources disponibles en forêt sont plutôt intéressantes : la France dispose de 15 millions d'hectares de forêt productive, le potentiel énergétique issu du bois semble très important.

L'évaluation de ce potentiel passe ainsi par une estimation des volumes, des biomasses et des stocks de carbone. Ces estimations sont basées principalement sur les données de l'Inventaire Forestier National (IFN), qui fournit les volumes de *bois-mort tige*.

En effet, pour estimer la ressource volumique en bois-énergie, une connaissance du *volume total* est nécessaire, or le passage du volume bois-mort au volume total n'est pas aisée : les modèles existants sont loin d'être universels.

L'objectif du projet EMERGE est donc d'élaborer des méthodes cohérentes, génériques, et robustes pour évaluer le potentiel du bois-énergie disponible en France (EMERGE pour Elaboration de Modèles pour une Estimation Robuste et Générique du bois-Energie).

EMERGE regroupe ainsi plusieurs acteurs et organismes de recherche compétents dans le domaine forestier et dans l'élaboration de modèles mathématiques :

- L'Office National des Forêts (ONF).
- L'Institut National pour la Recherche Agronomique (INRA) via le Laboratoire d'Etude des Ressources Forêt-Bois (LERFOB), et le laboratoire Biogéochimie des Ecosystèmes Forestiers (BEF).
- L'unité de recherche Ecosystèmes Forestiers du CEMAGREF (Institut de Recherche pour l'Ingénierie de l'Agriculture et de l'Environnement).
- L'Inventaire Forestier National (IFN).
- Le FCBA (Forêt, Cellulose, Bois-construction, Ameublement).
- Le Centre National Professionnel de la Propriété Forestière (CNPPF)
- Le Centre de coopération Internationale en Recherche Agronomique pour le Développement (CIRAD)
- Le Laboratoire franco-chinois en Informatique, Automatique et Mathématiques Appliquées (LIAMA).

2) L'Office National des Forêts

L'Office National des Forêts est un Etablissement Public à caractère Industriel et Commercial (EPIC), créé en 1966, qui dépend du ministère de l'Agriculture ainsi que de celui de l'Ecologie et de l'Aménagement du territoire. Son rôle principal est celui de gérer de façon durable les forêts publiques (à savoir les forêts domaniales et communales), qui représentent 4,4 millions d'hectares : c'est la sylviculture.

Cette gestion a un objectif principal : la production de bois que l'Office commercialise, les produits de vente des forêts domaniales entrant directement dans son chiffre d'affaires (42% en 2007).

Les autres activités de l'ONF sont :

- La surveillance de la forêt publique.
- Les baux de chasse en forêt domaniale.
- Les travaux effectués par le personnel de l'ONF pour les forêts communales, et facturés aux communes concernées.
- La location de concessions en forêt domaniale.

Enfin, l'ONF effectue l'accueil du public en forêt, afin de sensibiliser les personnes à la forêt.

Au sein de l'ONF, le département R&D est chargé de faire progresser les techniques et connaissances nécessaires à la gestion des forêts publiques. Ainsi, par des projets de recherche en propre ou en partenariat avec d'autres organismes de R&D, ce département vise à mettre au point de nouveaux *itinéraires sylvicoles*, à prendre en compte les changements climatiques dans la gestion, à faire une analyse prospective des nouvelles technologies disponibles, et le cas échéant d'organiser leur mise en production.

Au sein du pôle R&D, le site de Nancy, où s'est déroulé mon stage, est spécialisé dans la sylviculture de la hêtraie continentale, l'adaptation de la gestion forestière aux changements climatiques, et les méthodes d'inventaire. C'est dans ce cadre de l'étude des méthodes d'inventaire que l'équipe de Nancy est chargée de l'action 4.4 du projet EMERGE, qui vise à évaluer le potentiel du scanner laser terrestre pour des applications d'inventaire forestier.

3) Problématique du stage.

L'objectif du stage est ainsi d'évaluer le potentiel du scanner laser terrestre pour des applications d'inventaire forestier.

Comme nous le verrons par la suite, les données de base relevées lors d'un inventaire forestier sont le diamètre, la localisation (par rapport au centre de la placette) et l'essence des tiges concernées par cet inventaire. Nous avons ainsi cherché à évaluer les capacités de l'outil à reconstruire des diamètres et à retrouver la position des arbres, la détection d'essences étant un domaine qui nécessite d'autres compétences.

Il s'agit également d'évaluer la précision de l'outil en fonction de la distance et du diamètre

inventorié.

L'objectif est donc de mettre en place une chaîne de traitement informatique capable de nous permettre d'évaluer l'intérêt du scanner laser terrestre en foresterie.

II. Etat de l'art.

1) Inventaires forestiers.

a) Objectif

L'aménagement est un document qui planifie la gestion d'un massif forestier sur le long terme. Il contient, entre autres, les récoltes à prévoir ainsi que les opérations sylvicoles à mener sur ce massif afin d'aboutir au résultat souhaité.

L'établissement de ce document nécessite une phase d'étude préalable de l'état actuel de la forêt : on y effectue ainsi un inventaire. Il permet non seulement d'évaluer le *capital sur pied* du massif mais surtout d'effectuer un classement des parcelles, classement qui permettra de déterminer la manière dont la parcelle va être gérée (*régénération* du peuplement, mise en place d'un *peuplement irrégulier*).

Il existe deux méthodes d'inventaire : l'inventaire dit en plein, qui met en œuvre un relevé complet des tiges présentes sur une parcelle, ainsi que des inventaires statistiques. Pour évaluer le capital statistiquement, on crée sur cette forêt un maillage qui permet d'évaluer au mieux le peuplement. En chacun des nœuds de cette maille, on utilise deux types d'inventaire :

- La méthode des placettes, repérées par un maillage en forêt (voir annexe I).
- La méthode relascopique qui utilise le relascope de Bitterlich (voir annexe II).

Parmi la méthode des placettes, certaines d'entre elles sont réutilisées de manière fréquente afin de pouvoir estimer l'évolution de la forêt : ce sont les placettes permanentes. Ainsi, en dessinant une carte de la placette et en relevant certaines données comme un diamètre, nous sommes capables d'estimer la croissance annuelle des arbres, le pourcentage d'arbres morts, de souches...et donc d'avoir une vision à moyen terme et d'établir un modèle d'évolution de la forêt.

b) Données obtenues.

Les mesures chiffrées effectuées sur les placettes permanentes sont de 5 types :

- Mesure de diamètre, par l'intermédiaire d'un compas forestier.
- Mesure de circonférence au mètre ruban (si pas de mesure de diamètre).
- Mesure de distance, au décamètre ou par une méthode par ultrasons (avec un dendromètre, communément appelé Vertex, appareil de référence commercialisé par la société suédoise Haglöf).
- Mesure d'azimut, effectuée à la boussole.

- Détermination de l'essence, effectuée par l'agent.

On peut remarquer que les mesures de l'azimut et de la distance ne sont prises que pour pouvoir retrouver la placette. En effet, les placettes permanentes ne sont jamais repérées sur le terrain.

Enfin, les données qualitatives ne sont jamais relevées dans le cas d'une mesure « standard » effectuée dans un inventaire : tout dépend du protocole de mesure mis en place.

c) Méthodes de mesure.

Diamètre, circonférence :

L'estimation du diamètre se fait à 1,30 m de hauteur, à hauteur de poitrine (d'où l'appellation *Diameter Breast Height* ou DBH en anglais). L'agent positionne son compas forestier (ou son mètre ruban) à cette hauteur et mesure l'une ou l'autre des données.

Distance :

Elle s'effectue entre le centre de la placette et l'arbre considéré, mesurée au sol lorsque l'on utilise un décimètre, ou bien avec une mire, posée à 1,30 m de hauteur lorsque l'on utilise un dendromètre électronique.

Azimut :

Il se mesure également au centre de la placette par visée optique, par l'intermédiaire d'une boussole. On vise l'axe de la tige mesurée.

d) Limites actuelles.

Les limites de ces mesures effectuées sur les placettes permanentes sont principalement dues au type d'inventaire.

L'ONF utilise sur les placettes permanentes, deux types d'inventaires pour l'estimation des peuplements : l'inventaire à angle fixe et l'inventaire à surface fixe.

- L'inventaire à surface fixe consiste à obtenir des données sur l'intégralité des tiges présentes sur une surface prédéfinie (généralement un cercle de 10 m de rayon).
- L'inventaire à angle fixe utilise le relascope de Bitterlich, qui permet de conserver ou non un arbre dans l'inventaire, selon le rapport qu'il existe entre sa distance au centre de la placette et son diamètre (voir l'explication en Annexe II).

Ces deux types d'inventaires ont une limite basse de diamètre de relevé : 17,5 cm de diamètre mesuré à une hauteur d'un mètre trente, qui représente le diamètre minimal du sous-étage (perches et taillis).

Peuplement	Non précomptable				Précomptable									
Diamètres à 1,30 m	7,5 cm	10	15	17,5 cm	20	25	27,5 cm	30	35	40	47,5 cm	45	50	...
Essences arborescentes	Régénération	Perches et Petits Bois d'avenir												
		Sous-étage				PB			BM		GB et TGB			
Essences arbustives	Souille	taillis et Perches												

Figure 1: Classes de diamètre des arbres inventoriés

Chacun de ces inventaires présente ses avantages et ses défauts :

- L'inventaire à angle fixe permet d'évaluer globalement l'état du peuplement, comme il implique la mesure de la surface terrière de la placette (qui sera ramenée à l'hectare). Par conséquent, le décompte des grosses tiges est favorisé par rapport à celui des petites tiges.
- L'inventaire à surface fixe nous permet d'évaluer l'ensemble des tiges sur une surface donnée, ce qui implique donc une évaluation plus importante des petites tiges par rapport aux plus grosses. Enfin, cet inventaire « exhaustif » peut également comporter des oublis.

Tableau 1: Forces et faiblesses des différents types d'inventaire.

	Surface fixe	Angle fixe
+	Données complètes sur 10 m.	Etat du peuplement
-	Privilège les petites tiges, oublis	Privilège les grosses tiges

2) Le LiDAR (Light Detection And Ranging)

a) Présentation de l'outil.

Le LIDAR est un outil optique de télédétection qui mesure les propriétés de la lumière dispersée. Si son appellation est très proche de celle du RADAR (Radio Detection And Ranging), c'est parce que ces deux technologies ont un principe de fonctionnement similaire : pour déterminer une distance, le LIDAR analyse les ondes émises par la source et réfléchies par l'obstacle et mesure la durée entre l'émission et la réception de l'onde.

La différence entre les deux technologies est que le RADAR utilise les ondes radio, de grandes longueurs d'ondes, alors que le LIDAR utilise des fréquences plus élevées (donc des longueurs d'onde

plus faibles), comme les ondes électromagnétiques du domaine visible, de l'ultraviolet et de l'infrarouge proche.

Comme les ondes électromagnétiques sont réfléchies par un objet lorsqu'une discontinuité diélectrique apparaît sur sa surface, les matériaux conducteurs d'électricité sont très bien réfléchis, ce qui implique donc que les matériaux non ou très peu conducteurs, comme le bois ou la roche sont très peu réfléchis.

Le laser est une source lumineuse cohérente et dense, qui émet des rayons à longueurs d'onde bien plus faibles que celle des ondes radio (100 nm contre 0,1 mm au moins). Or à ces faibles longueurs d'onde, les rayons sont bien réfléchis par les surfaces faiblement conductrices (on parlera de rétrodiffusion plutôt que de réflexion), la technologie est donc adaptable à de nombreux cas. De plus, le faisceau lumineux émis par le laser est très étroit, il est donc possible d'effectuer une mesure de distance à très haute résolution

b) Utilisation dans l'industrie et la recherche.

Du fait des résolutions envisageables et de l'utilisation possible sur des matériaux très divers, les applications du LIDAR sont très nombreuses :

- Le LIDAR a été appliqué dans un premier temps à l'étude des nuages et de la composition atmosphérique. En effet, les particules présentes dans l'atmosphère ont une taille proche du micromètre (10^{-6} m), alors que les rayons émis par le LIDAR ont une longueur d'onde comprise entre 600 et 1000 nm (soit $0,6 \cdot 10^{-6}$ m et 10^{-6} m), ce qui fait que les particules réfléchissent très bien la lumière issue du laser.
Il existe différents types de LIDAR dits atmosphériques, qui ont des applications variées, comme la concentration en particules de nuages et d'aérosols, concentration en gaz de l'atmosphère, mesure de la vitesse du vent, via effet Doppler, etc.
- Le LIDAR a été utilisé ensuite en astronomie : la mission Apollo 11 a déposé sur la surface de la Lune des miroirs réflecteurs, ce qui a permis depuis de mesurer la distance Terre-Lune, et d'aboutir à de nombreux résultats concernant l'évolution de la trajectoire de la Lune.
Depuis cette application spatiale réussie, et compte-tenu des résultats obtenus dans l'étude de l'atmosphère, le LIDAR est utilisé dans les sondes spatiales pour pouvoir mesurer la composition d'atmosphères.
- Le LIDAR pouvant effectuer des mesures précises sur de grandes distances, il est désormais utilisé pour effectuer des *MNT* (Modèle Numérique de Terrain). Il est ainsi utilisé pour établir des cartes d'élévation de terrain, des voies de communication, etc.
Enfin, dans le cadre de travaux de recherche, il est possible de mesurer les variations de la hauteur du terrain lors de tremblements de terre (géologie) ou encore d'étudier d'anciennes voies de communication (archéologie) [1].

- Le LIDAR pouvant également mesurer rapidement et précisément les distances, il est utilisé en tant qu'évaluateur de distance dans l'automobile. Lorsqu'il est bouclé au moteur, il joue le rôle de capteur, et permet à l'ordinateur de bord de ralentir le véhicule si ce dernier est trop près de celui qui le précède. Les perspectives à très long terme sont l'automatisation de la circulation.
- La dernière application industrielle du LIDAR est la numérisation 3D de scènes : il permet d'obtenir des cartes en trois dimensions d'un paysage. On parle ici de Terrestrial Laser Scanner (TLS), ou scanner laser terrestre.

c) Le scanner laser terrestre.

Compte tenu des caractéristiques très intéressantes du TLS (haute résolution, grande précision), ce produit s'est développé dans la gamme des appareils de contrôle non destructifs. Il est désormais utilisé dans une très grande majorité d'industries de haute technicité :

- Les industries aérospatiales, aéronautiques, ferroviaires et automobiles utilisent cette technologie pour inspecter et contrôler l'alignement des pièces lors de la fabrication ou du montage. Cela permet, en cas de problème observé, de le corriger rapidement. Les industries lourdes sont également consommatrices de cet outil, qui permet un contrôle qualité directement sur la chaîne de production, ce qui évite le déplacement de matériel très lourd et donc une perte de temps conséquente.
- La transformation de scène réelle en scène numérique est très utile dans le domaine de la rétro-conception (manque de données CAO, scène fugace) : le TLS est ainsi utilisé dans les domaines de l'ingénierie inverse (pour reproduire ce qui existe déjà) ou encore en criminologie (reconstruction numérique d'une scène de crime).
- Les structures naturelles ou civiles s'altérant au cours du temps, il est nécessaire, en cas de travaux à effectuer, de connaître l'état actuel de cette structure, ce qui permet d'agir en conséquence.

Ces applications étant très variées et exigeantes en termes de performance, leur développement a permis d'améliorer les caractéristiques du produit. Les performances standard actuelles des produits fabriqués par les grands constructeurs de TLS (Leica, Faro) sont actuellement :

- o Résolution : 8000x3500 points
- o Portée min/max : 1m/120m
- o Erreur à 25m : 2mm
- o 360° horizontal / 320° vertical
- o Puissance du laser : 20mW

Enfin, le rayonnement utilisé par le TLS est situé dans le domaine du visible lointain et de l'infrarouge proche (longueur d'onde comprise entre 620 et 800 nm).

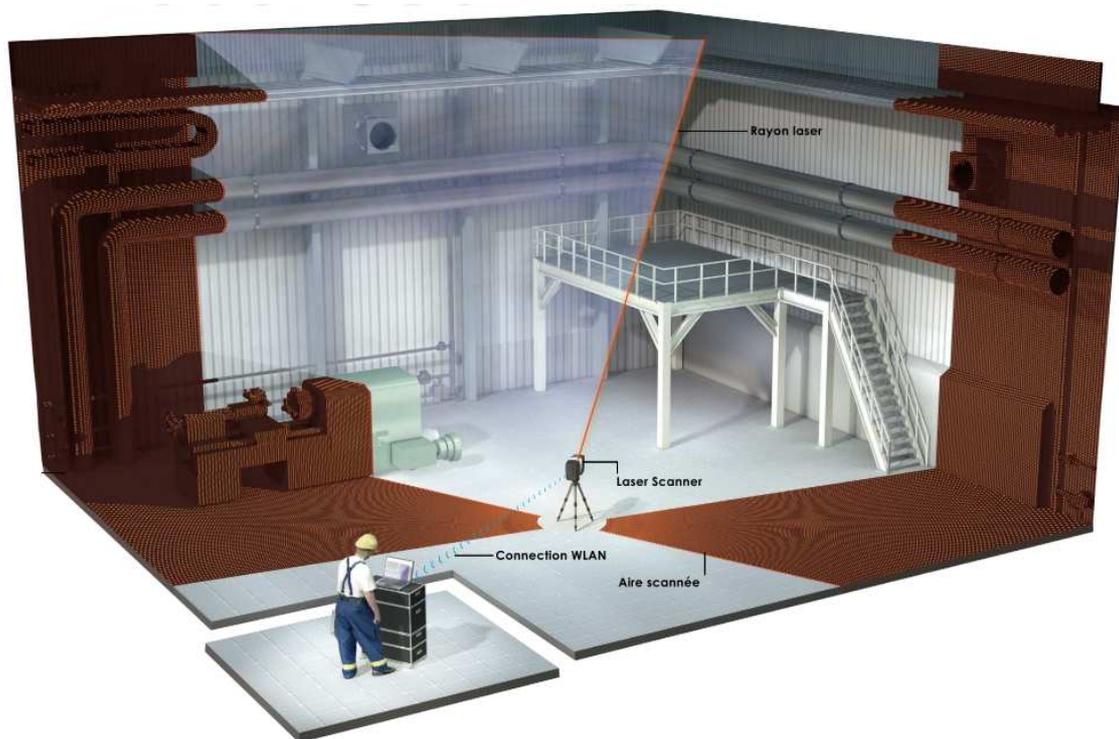


Figure 2: Principe de fonctionnement du TLS (issu de la documentation FARO "Laser Scanning infographic")

d) Application dans le domaine forestier.

Le LIDAR appliqué à la foresterie utilise les techniques précédemment évoquées, à savoir l'édition de MNT et la numérisation tridimensionnelle de scènes.

A l'heure actuelle, la cartographie de la forêt via un LIDAR installé sur un avion est de loin l'application forestière la plus développée. La Suisse et le Canada utilisent déjà le LiDAR « aérien » en tant qu'outil de gestion courante : en effet, la technique d'édition du MNT peut être quelque peu modifiée pour obtenir un Modèle Numérique de Canopée. Ce dernier, qui est soustrait au MNT préalablement édité, nous permet d'obtenir une carte de la hauteur réelle de la canopée et donc la hauteur des arbres.

La recherche sur l'application du LiDAR n'en est qu'à ses débuts en France, compte-tenu du fait que les peuplements forestiers français sont beaucoup plus diversifiés qu'en Suisse ou au Canada, que ce soit en termes d'essences que de type de peuplement. Les résultats existant à l'étranger ne sont donc pas applicables de façon universelle en France, un travail préalable de calibration s'avère nécessaire.

La numérisation 3D de scènes forestières en est à ses prémices. Les recherches déjà effectuées à l'étranger montrent qu'il est possible, avec cet outil, de détecter des arbres et de reconstruire les diamètres des tiges [2]. Cependant, ces résultats ne sont pas transposables à la forêt française sans étude préalable : non seulement les résultats ont été obtenus sur des peuplements majoritairement

composés d'épicéas, d'âge supérieur à 30 ans, mais surtout aucun algorithme utilisé n'a été publié (travail en collaboration avec Treematrix).

3) Le traitement des données LiDAR.

Compte-tenu des diverses applications autres que forestières du TLS, on peut ainsi observer aisément l'intérêt de travailler avec des données de ce type : non seulement l'outil permet de conserver la mesure mais surtout de la standardiser, aucun facteur humain n'intervenant dans cette dernière.

De plus, il n'y a pas de facteur lié au terrain ni à la méthodologie de l'inventaire, l'outil devrait donc nous permettre d'aller au-delà de ces problèmes de méthode, en nous permettant d'obtenir les données caractéristiques de chacun des inventaires (inventaire exhaustif sur une grande surface, capital sur pied) mais aussi d'obtenir des données annexes (MNT local sur la scène, mesure de diamètres à hauteur variable pour nous donner le volume de l'arbre, le houppier, la hauteur totale, etc.).

III. Matériel et méthodes.

A) Matériel et participants au projet.

1) Caractéristique du TLS (Terrestrial Laser Scanner).

Le TLS utilisé dans le cadre du projet est le Laser Scanner Photon 120 de la société FARO, dont les caractéristiques techniques sont données dans le tableau suivant :

Tableau 2: Caractéristiques techniques du TLS Faro Photon 120

Portée du signal	120 m
Erreur linéaire à 25m et à 84% de réflectance	2 mm
Résolution usuelle	8000x3500
Durée usuelle	233 s
Résolution maximale	470000x16384
Vitesse de mesure	jusque 976k pixels/s
Puissance du laser	20 mW
Longueur d'onde	785 nm
Angle de champ vertical	320°
Angle de champ horizontal	360°
Poids	14,5 kg

L'appareil est quant à lui disposé sur un trépied en fibre de carbone, qui comprend également un niveau à 3 bulles. Le scanner est par ailleurs équipé d'un inclinomètre qui corrige automatiquement le résultat final (précision 0,02°, résolution 0,001°, portée ±15°).

Le scanner émettant un faisceau laser, peu énergétique mais toutefois dangereux pour les yeux, un outil de commande à distance Wifi a été fourni.

Le scanner fournit par ailleurs différentes données sur les points qu'il relève :

- Coordonnées cartésiennes X,Y,Z de chacun des points relevés.
- Coordonnées sphériques R, θ , ϕ de chacun des points relevés.
- Réflectance de chacun des points relevés.

Enfin, le scanner utilisé durant ce stage a été est le même scanner que celui qui a été acquis par l'INRA de Champenoux dans le cadre du projet EMERGE.

2) Outils informatiques utilisés.

a) Poste de travail.

L'objectif du projet étant de mettre au point un outil informatique pour valoriser les mesures acquises précédemment, une station de travail a été spécialement achetée, et possède les caractéristiques suivantes :

HP Z800 Workstation

Processeur Intel Xeon X5550 cadencé à 2,67GHz.

32 GB de mémoire vive disponible.

Carte graphique NVIDIA Quadro FX 1800, cadencée à 768 Mo de mémoire.

Windows XP Service Pack 2, version 64 bits.

Ces caractéristiques permettent un traitement des données de gros volume, ainsi que l'utilisation parallèle de plusieurs logiciels. Enfin, le système d'exploitation 64 bits permet d'exploiter pleinement la quantité de mémoire vive disponible.

b) Logiciels utilisés.

Logiciels Faro.

Afin de pouvoir extraire et observer les scans issus du scanner laser, l'utilisation des logiciels Faro s'est avérée nécessaire.

En effet, l'acquisition du TLS par l'INRA a également conduit à l'achat d'une licence du logiciel **Faro Scene**, qui permet de visualiser la scène en vue panoramique, en vue 3D, d'extraire les jeux de points acquis en plusieurs formats (Ascii, binaire, VRML, DXF, etc.) de repérer des sphères de recalage disposées sur une scène (pour fusionner plusieurs scans) et de distinguer des plans, des cylindres, etc.

Langages de programmation.

Les développements ont débuté en **Java**, langage de prédilection de M. Michael Krebs. Il présente l'avantage d'être utilisable sous la majorité des grands systèmes d'exploitation actuels (Windows, Linux, Mac), mais a l'inconvénient d'être peu performant avec des jeux de données de grande taille, et ce malgré les précautions prises pour exploiter au mieux la mémoire vive disponible.

La version finale du logiciel a été développée sous le langage **C++** dans sa version **Qt**, langage performant et utilisé dans la très grande majorité des applications Windows.

Enfin, les tests ont quant à eux été écrits en langage proche du **Script**, le langage du logiciel R (cf « Logiciel de calcul statistique »), qui présente une interface utilisateur en propre, ce qui nous permet de nous affranchir de sa création.

Environnements de développement.

L'environnement de développement Java utilisé est **Eclipse**, qui est un logiciel open source et leader mondial sur le marché des environnements Java.

La bibliothèque de classes C++ utilisée est **Qt**, mis au point par Nokia et disponible en une version légale et gratuite.

Logiciel de calcul statistique.

N'étant pas programmeur, il m'a fallu, afin de pouvoir tester rapidement des jeux de données, utiliser un langage et un logiciel plus facile et rapide d'accès que ceux précédemment cités.

Le logiciel **R** est un logiciel de statistique couramment utilisé au sein du service Recherche & Développement de l'ONF, qui présente de nombreux avantages :

- C'est un logiciel open source.
- L'apprentissage du langage est très rapide et aisée.
- Il permet d'utiliser des données externes très simplement.
- Il est utilisable dans les grands systèmes d'exploitation (Windows, Unix, Linux, Mac).
- Il dispose d'une interface utilisateur en propre.
- Enfin, R existe désormais en version 64 bits, ce qui permet d'utiliser au mieux les capacités des processeurs multi-cœur et des fortes capacités de mémoire vive disponibles sur la station de travail mise à disposition pour ce stage.

Enfin, R privilégiant les résolutions numériques au calcul formel, il a parfois été nécessaire d'effectuer certains calculs sous un logiciel de calcul formel, à savoir **Maple**. Cela a permis de pouvoir comparer les résultats théoriques aux résultats numériques calculés sous R.

3) Les partenaires externes au projet.

Le centre Arts et Métiers ParisTech de Cluny.

C'est le partenaire le plus important, via l'implication pendant plus de la moitié de ce projet de deux membres du laboratoire LaBoMaP du centre de Cluny :

- M. Michael KREBS, développeur du logiciel Computree (voir la suite).
- M. Laurent BLERON, qui a fourni d'importants conseils sur les différents traitements mathématiques mis en place.

Le laboratoire Lerfob de l'INRA de Champenoux.

Le laboratoire ayant travaillé et œuvrant actuellement sur la thématique du TLS en foresterie, nous sommes bien évidemment tournés vers ce partenaire du projet EMERGE, par l'intermédiaire de deux de ses collaborateurs :

- M. Mathieu DASSOT, préparant la thèse de doctorat « Biomécanique de l'arbre et applications du LIDAR terrestre » au Lerfob, pour sa connaissance de l'outil.
- M. Thiéry CONSTANT, chargé de recherche au Lerfob, pour la mise en place du jeu de validation.

4) Mise en œuvre d'un jeu de validation en forêt de Bezange.

a) Objectif du jeu de validation.

Après avoir mis au point notre base algorithmique, lors de l'évolution de la phase de tests, nous avons remarqué que les jeux de données disponibles étaient issus d'inventaires à angle fixe. Or, compte-tenu de notre volonté d'évaluer le TLS comme outil de réalisation d'inventaires les plus exhaustifs possibles, il nous est apparu souhaitable d'obtenir un jeu de données le plus complet possible. Nous avons donc cherché à effectuer un ou plusieurs inventaires à surface fixe, afin d'évaluer :

- L'exhaustivité de détection apportée par le TLS (l'inventaire se fera donc sur la plus grande surface possible, donc pas limitée à un cercle de 10 m de rayon).
- La précision sur la détermination des diamètres.
- La précision sur la localisation des tiges.
- L'influence du diamètre recherché, de sa distance par rapport au scanner et des essences (non traité dans le cadre de ce stage).

b) Choix du massif forestier de Bezange.

Il a donc été nécessaire de rechercher un massif forestier :

- Public (donc une forêt communale ou domaniale).
- Proche de l'INRA de Champenoux et de l'ONF de Velaine-en-Haye.

- Présentant un peuplement varié en termes de densités locales, d'essences et de diamètres de tige.
- Disposant d'un réseau de placettes permanentes, afin de pouvoir comparer un relevé antérieur aux futurs relevés.

Compte-tenu de ces conditions et après avoir discuté avec Olivier LANTER (aménagiste de la forêt de Bezange et présent sur le site de Velaine-en-Haye), il a été décidé de travailler en forêt domaniale de Bezange-la-Grande (Meurthe-et-Moselle).

c) Choix des placettes.

Après avoir obtenu une carte de la forêt (contenant la délimitation des parcelles et le réseau des placettes permanentes, voir annexe III) et après accord de M. Stéphane COUBRONNE (agent patrimonial de la forêt de Bezange), il a été nécessaire de passer par une étape de prospection sur le terrain, afin de déterminer quels seraient les sites les plus favorables à nos mesures.

Afin de ne pas partir tête baissée en forêt, il nous a été mis à disposition par M. Frédéric BEDEL (responsable des aménagements en Meurthe-et-Moselle) la base de données contenant les cartes des placettes permanentes telles qu'elles ont été relevées en août 2006.

La forêt de Bezange présente la caractéristique d'être séparée en termes de gestion, à savoir qu'une grande majorité de sa superficie est exploitée (les opérations sylvicoles y sont menées), et qu'une autre est une Réserve Biologique Intégrale (*ou* RBI), où les opérations sylvicoles sont interdites. Cette zone présente donc les différents stades possibles d'évolution des tiges, elle est donc hétérogène, ce qui est intéressant dans l'objectif de l'étude d'un peuplement varié.

Nous avons donc décidé de travailler dans un premier temps dans la RBI. Ainsi, nous avons pu isoler 8 placettes permanentes rentrant dans les caractéristiques précédemment établies.

Cependant la tempête Xynthia, qui a traversé la France entre le 26 et le 1^{er} mars, a eu des effets dévastateurs sur ces parcelles situées dans la RBI : elles ont toutes été touchées par la tempête et nombre d'arbres ont été déracinés. Les placettes choisies ont ainsi toutes été touchées par la tempête, ce qui fait qu'aucune mesure satisfaisante n'aurait été possible.

Après une nouvelle discussion avec Stéphane COUBRONNE et Olivier LANTER, nous nous sommes réorientés vers des parcelles plus classiques, mais où les récoltes n'étaient pas à l'ordre du jour, pour éviter d'influencer le travail des agents sur le terrain.

C'est pourquoi nous avons donc choisi les placettes 221 et 222 appartenant respectivement aux parcelles 4 et 1. Elles présentent toutes les deux un nombre élevé d'essences différentes (au moins 8), une densité locale intra-placette variable et des diamètres très variables selon les distances.

d) Mesures effectuées.

Le choix des placettes ayant été effectué, nous avons mis en place le protocole d'expérience (disponible en annexe IV).

L'objectif étant de réaliser un inventaire le plus exhaustif possible, nous avons relevé les données suivantes :

- Localisation en coordonnées polaires par rapport au centre de la placette.
- Essences.
- Circonférences, diamètres dans puis orthogonal à l'axe de visée, et ce à une hauteur fixe.

Le choix d'un relevé à hauteur fixe vient de l'évolution des traitements qui ont pu être effectués : nous avons en effet mis en place une extraction des cercles à hauteur nulle par rapport au scanner pour chaque arbre reconstruit, ainsi, un relevé à hauteur fixe des diamètres rend les traitements futurs plus rapides et faciles (voir la suite).

Pour repérer cette hauteur fixe sur la placette, nous avons utilisé une *station totale*. Afin de pouvoir distinguer cette hauteur sur le scan sous les logiciels Faro, nous avons l'avons représenté par deux traits de peinture.

Pour étudier plus particulièrement les performances en termes d'obtention de points du scanner laser, nous avons marqué l'arc de cercle visible depuis le centre de la placette de quelques arbres, pour pouvoir le comparer à ce que l'on obtient en sortie de traitement.

Enfin, nous avons fait fonctionner le scanner sur ces deux placettes, pour pouvoir comparer les mesures terrain aux données issues du scanner.

B) Développement du logiciel COMPUTREE.

Le développement du logiciel COMPUTREE s'est fait en plusieurs étapes, ses évolutions étant liées aux diverses évaluations qui ont été effectués au fur et à mesure du développement.

1) Premières analyses des jeux de données.

Avant de commencer à tenter de programmer, nous nous sommes tournés vers l'analyse des données en notre possession :

- Un arbre, au premier abord, est un objet cylindrique dont le diamètre varie lorsqu'on le parcourt sur toute sa hauteur.
- Lorsque l'on observe par l'intermédiaire de la vue 3D des logiciels Faro un nuage de points, on peut noter que les points, pour une hauteur donnée, représentent globalement un arc de cercle.

- Ce qui distingue un arbre du sol est que l'arbre est un objet qui croît en coordonnées verticales.
- Cette même vue 3D nous fait également remarquer qu'une grande majorité des points relevés par le TLS sont des points du sol.

Un arbre est un cylindre dont le diamètre varie.

On pourra modéliser un arbre par un cylindre à diamètre variable, variation qu'il faudrait pouvoir encadrer lorsque l'on passe d'une hauteur à une autre.

Une tranche d'arbre est représentée par un ensemble de points en forme d'arc de cercle.

Pour pouvoir retrouver les diamètres (ou du moins ce qui s'en rapproche le plus pour un arbre), il faudra utiliser des algorithmes de reconstruction de cercles ou d'ellipses.

Un arbre est un objet qui croît verticalement.

Les deux remarques précédentes ainsi que le constat simpliste qu'un arbre est un objet qui est orienté verticalement fait que l'on peut modéliser un arbre comme un enchaînement vertical de cercles ou consorts.

Le TLS relève surtout des points du sol.

Le sol risque de poser des problèmes pour la reconstruction des arbres. Il faudra peut-être chercher à le discriminer avant de reconstruire les tiges.

2) Algorithmes mis en œuvre.

Ces algorithmes ont été codés dans un premier temps en langage JAVA.

a) Séparation du nuage en tranches.

Si l'on peut modéliser un arbre comme un enchaînement vertical de formes, cela signifie qu'il faut être capable de distinguer des formes sur plusieurs tranches du nuage de points, et de préférence localisées au même endroit.

En observant des tranches du nuage que nous avons à disposition, nous avons pu noter que les mêmes motifs se répétaient. Ces motifs sont localisés au même endroit, et ce tant que l'on travaille avec des tranches contiguës d'épaisseur fine.

En effet, lorsque l'on observe des tranches d'une épaisseur proche de cinq centimètres, on observe nettement sur une vue plane que cette tranche (qui est donc projetée) ne représente plus un arc de cercle ou une portion d'ellipse mais une succession d'arcs de trajectoires quasi-parallèles et non concentriques.

Ainsi, les algorithmes de reconstruction de forme ne permettront jamais de reconstituer la meilleure forme : ils ne nous donneront que la meilleure approximation d'un ensemble fini de formes, ce qui ajoute un effet « moyenne » qui n'est pas sans jouer sur la précision du résultat final.

Dans un second temps, on peut noter que sur deux tranches séparées de plusieurs dizaines de centimètres, il se peut que les motifs que l'on souhaite détecter se déplacent d'autant sur le plan horizontal (les conséquences seront évoquées dans le point « Reconstruction des arbres »).

On peut tirer de ces observations le résultat suivant : il sera préférable de travailler avec des tranches de nuage les plus fines possibles, qui nous permettront de limiter l'effet de projection précédemment évoqué, tout en ayant une épaisseur suffisamment importante pour pouvoir observer un motif cohérent.

Cet algorithme a ainsi été mis en œuvre initialement dans un logiciel dédié :



Figure 3: Logiciel de tri en couches (format d'entrée *.xyb (fichier binaire), format de sortie est le *.xybs)

Enfin, le fait de travailler sur des couches à épaisseur constante permet d'éviter l'effet du sol : en effet, les placettes que nous serons menés à étudier présentent une topographie stable (les pentes sont faibles et il n'y a pas de petites collines), ce qui fait que le sol sera ainsi concentré sur les couches les plus basses. Il sera donc aisé d'en supprimer ou d'en compenser les effets.

b) Détermination des formes.

Il est donc désormais nécessaire de pouvoir reconstruire les motifs observés sur ces tranches. Nous nous sommes ainsi tournés vers la détermination de cercles et d'ellipses.

En ce qui concerne le cercle, il existe de nombreuses méthodes numériques de reconstruction possibles. La résolution en termes des moindres carrés (appelé dans la suite du rapport *RMSE*) est la plus aisée à mettre en œuvre. Il en existe d'autres, et en particulier la méthode du maximum de vraisemblance (*MLE*), qui est cependant plus complexe à mettre en œuvre (les algorithmes ne sont plus sous forme de calcul direct mais plutôt de matrices à inverser).

Il en va de même pour la reconstruction d'ellipses : Chernov et Taubin [3] ont créé deux algorithmes qui permettent de reconstruire des ellipses dans un plan, et qui nous renvoient les 6 coefficients de l'équation de l'ellipse (rappel : $ax^2 + by^2 + cxy + dx + ey + f = 0$).

Ainsi, afin d'évaluer la légitimité d'une reconstruction sous la forme de cercles ou d'ellipses, il a donc été choisi d'utiliser les algorithmes les plus simples pour pouvoir reconstruire des cercles et des ellipses, à savoir la *RMSE* et la reconstruction elliptique de Chernov.

c) Reconstruction des cercles et des ellipses.

Obtenir un cercle et une ellipse à partir d'un jeu de points quelconque est une tâche aisée, mais malheureusement très peu représentative de la réalité. En effet, les algorithmes qui existent vont toujours chercher à obtenir le meilleur cercle passant par l'ensemble des points d'une même couche, ce qui peut donner, pour une tranche d'un jeu de points de 20 m de diamètre, un diamètre ou un demi-grand axe de 20 m...

Il est donc nécessaire de pouvoir localiser et distinguer les formes voulues sur ces mêmes tranches. Des algorithmes sur le sujet existent déjà, et la transformée de Hough [4] (algorithme de reconnaissance de formes à l'origine des traitements d'images numériques) semble indiquée. En effet, elle est utilisée en particulier pour retrouver des cercles dans des jeux de points, elle est donc parfaitement applicable à notre cas.

Cependant, des essais préliminaires menés sur des tranches de scan montrent des limites quant à la résistance au bruit de la transformée de Hough. En effet, une tranche d'une placette de 20 tiges nous fait apparaître au moins deux fois plus de cercles que ce qui a pu être relevé lors de l'inventaire ou observé sur l'image panoramique issue des logiciels Faro, ce qui signifie que le bruit est détecté et donc reconstruit.

Il a donc été nécessaire d'utiliser une autre méthode pour être capable de distinguer la forme. Il a été choisi de mettre les points dans ce que l'on a nommé groupes, selon la méthode suivante :

- On prend un point de la tranche, point qui définit un groupe.
- On évalue la distance de ce point avec un autre point. Si celle-ci est inférieure à la distance maximale d'appartenance à un groupe prédéfinie, ce point appartient désormais au groupe. Si ce n'est pas le cas, on prend un autre point et on recommence. Le groupe a désormais deux points.
- On effectue le même test de distance entre un point et ces deux points, et on ajoute ou non un troisième point au groupe.
- On continue jusqu'à ce que l'ensemble de la tranche ait été traité.

La mise en groupe des points permet en réalité de s'approcher au mieux de la forme souhaitée : le barycentre du groupe en devient le centre.

On considère enfin que le groupe en est réellement un quand son nombre de points dépasse un seuil, que l'on peut fixer sur l'interface graphique du programme. En effet, en observant des nuages filtrés, on s'aperçoit que, pour une seule et même tranche, les points qui n'appartiennent pas à un arbre (et qui sont donc des points de bruit qui résistent aux filtres) sont généralement isolés et en très faible quantité. Ainsi, l'algorithme de groupage des points peut effectivement les grouper, mais ils seront finalement éliminés par ce filtre.

Ainsi, dès que la détection des groupes est effectuée, on peut chercher à déterminer les diamètres des cercles ou les deux axes des ellipses (donc son aire) pour chacune des couches.

Enfin, comme il l'a été précisé précédemment, les algorithmes que nous allons utiliser vont toujours rechercher à retrouver le meilleur cercle parmi une couche donnée. Ainsi, afin de limiter les erreurs

de détection, nous avons limité le rayon minimal et le rayon maximal de reconstruction de cercle (idem pour les aires des ellipses), afin d'obtenir des résultats cohérents avec ce que l'on peut s'attendre à retrouver au sein d'un peuplement.

d) Reconstruction des arbres.

Les formes étant connues pour chacune des tranches, il faut désormais arriver à les agglomérer pour pouvoir reconstruire les arbres. Pour cela, on utilise le fait qu'un arbre est un objet en Z croissant (ou décroissant) et que les coordonnées planes, pour deux hauteurs Z proches, ne varient que très peu.

Pour cela, on définit l'algorithme de la manière suivante :

- On part de la couche supérieure du nuage.
- On choisit une forme reconstruite, elle constitue un arbre.
- On regarde toutes les formes reconstruites de la couche inférieure : s'il y en a une qui est très proche de la première en distance plane, alors on l'ajoute à la catégorie arbre, sinon, on arrête là.
- On effectue le même travail pour la couche directement inférieure à celle que l'on vient d'évaluer, et ainsi de suite.

Ainsi, après les traitements successifs du jeu de données par les 4 algorithmes ici présentés, le seul bruit qui peut rester a été modélisé comme un cercle ou une ellipse. Or ce bruit est généralement très concentré en épaisseur (moins de 5 cm, très souvent dû aux branches parasites et aux hautes herbes). C'est donc la raison pour laquelle il a été défini un critère de minimum de cercles nécessaires à la reconstruction d'un arbre, qui permet d'éliminer encore une partie du bruit.

e) Visualisation des résultats.

Afin de pouvoir exploiter ces résultats et commencer le travail de comparaison, il faut pouvoir extraire et analyser le travail de l'algorithme.

En ce qui concerne l'extraction des données issues de l'algorithme, nous exportons deux tableaux de données :

- Le premier qui référence le nombre d'arbres reconstruits et le nombre de cercles par arbre.
- Le second qui nous donne l'ensemble des formes reconstruites, avec les coordonnées X,Y,Z, de leur centre, le rayon ou les deux demi axes, et l'erreur sur le rayon ou sur l'aire, selon la forme considérée.

La visualisation des résultats obtenus se fait via l'installation du package Java 3D. L'interface de visualisation mise en place permet d'observer le nuage de points et/ou d'observer les formes et les arbres reconstruits.

Enfin, l'interface utilisateur permet de modifier les paramètres « Nombre de points minimum pour faire un groupe », « Nombre de cercles minimum pour faire un arbre », « Rayon/aire minimale à observer », « Rayon/aire maximale à observer », « Erreur maximale en rayon/aire », « Distance maximale d'un point à un groupe ».

Le visuel de COMPUTREE est disponible en annexe V.

3) Evaluation des performances.

Cette évaluation s'est faite sur les premières versions du logiciel développé en Java.

Nous avons évalué les performances de COMPUTREE en deux parties : nous avons travaillé sur un jeu de données complet, afin d'observer le comportement du logiciel sur les données qu'il sera conduit à traiter, puis fait varier les différents paramètres de l'algorithme, en choisissant des arbres type, pour obtenir une « plage » de bon fonctionnement du logiciel.

a) Travail sur une placette permanente.

1) Caractéristiques de la placette test et des données traitées.

La placette choisie (nommée c13p5g) comporte 37 tiges, dont les circonférences relevées sur le terrain varient de 280 mm à 2000 mm. Leur distance au centre de la placette est comprise entre 3,5 m et 21 m, et deux essences sont représentées : le hêtre et le chêne sessile, avec une forte proportion de hêtre. Cette placette représente ce que l'on peut observer sur un réseau de placettes permanentes hêtre-chêne, d'où son choix.

Enfin, nous avons étudié les performances sur plusieurs plages de longueur :

- Entre 1 et 5 m de distance par rapport au centre de la placette.
- Entre 5 et 10 m.
- Entre 10 et 15 m.
- Entre 15 et 25 m.
- Entre 1 et 25 m (le nuage complet).

L'étude de différentes plages de longueur vient du fait que l'on pouvait penser gagner en temps de calcul en séparant les nuages.

2) Choix des paramètres.

La distance minimale d'un point à un groupe a été choisie de manière à limiter les effets d'ombre sur la détection. Comme on peut le voir sur le schéma suivant, les points situés dans le cône d'ombre de

la tige de faible diamètre sont inobservables. Les algorithmes pourraient alors distinguer deux tiges au lieu d'une seule pour le plus gros diamètre. On peut remarquer que ce phénomène intervient majoritairement pour les arbres les plus proches : en effet, pour des distances très élevées, les cônes d'ombre sont très étroits angulairement, le seuil maximal ne peut jamais être atteint. Ainsi, nous avons choisi un seuil moyen de 30 cm.

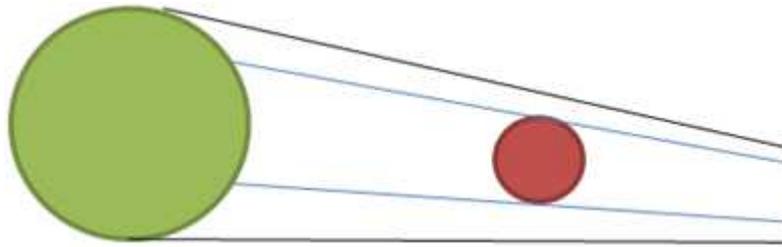


Figure 4: cône d'ombre

La précision recherchée sur le résultat final étant maximale, nous avons choisi de caler l'erreur maximale sur l'erreur de mesure du diamètre effectuée au compas forestier, à savoir 1 cm.

L'objectif étant de relever tous les arbres présents sur la scène observée, le diamètre minimal doit correspondre au diamètre de ce que l'on peut considérer sur le terrain comme une tige à part entière, à savoir 2 cm. Le diamètre maximal doit quant à lui être représentatif de ce que l'on peut trouver en forêt, nous avons ainsi choisi une valeur de 120 centimètres.

Le choix du nombre minimal de points nécessaires pour former un groupe s'est fait sur l'observation suivante : afin de reconstruire au mieux les formes, il faut un nombre de points suffisant à représenter. Nous avons donc choisi une valeur minimale de 10 points car, en-dessous de ce seuil, les résultats deviennent aberrants.

Enfin, les arbres numériques reconstruits correspondent parfois à du bruit (feuilles, hautes herbes, etc.) qui viennent perturber le résultat final. On remarque que ce bruit est généralement une accumulation d'une quantité très faible de cercles (moins de 20). Nous avons donc décidé de fixer le paramètre de nombre minimal de cercles nécessaires pour former un arbre à 20 cercles.

3) Observations et interprétations.

Ces observations concernent non seulement le temps de traitement mais aussi la reconstruction, d'un point de vue strictement visuel.

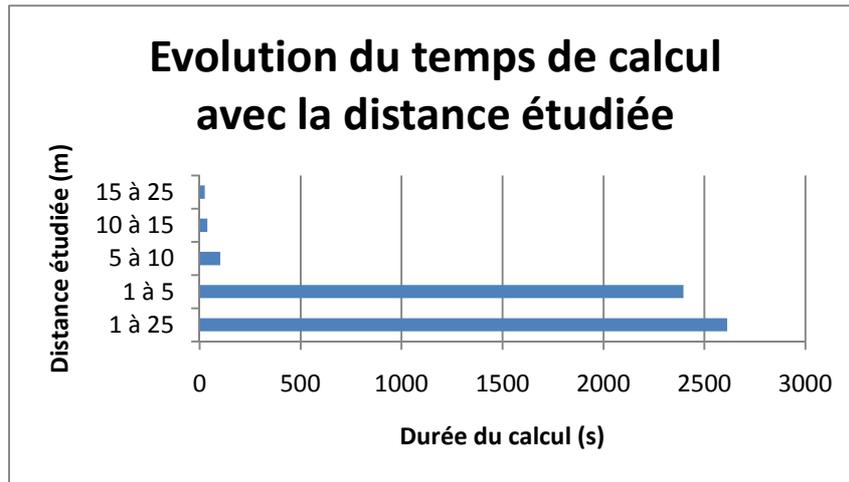


Figure 5: Evolution du temps de séparation en couches Z selon la fourchette de distance étudiée

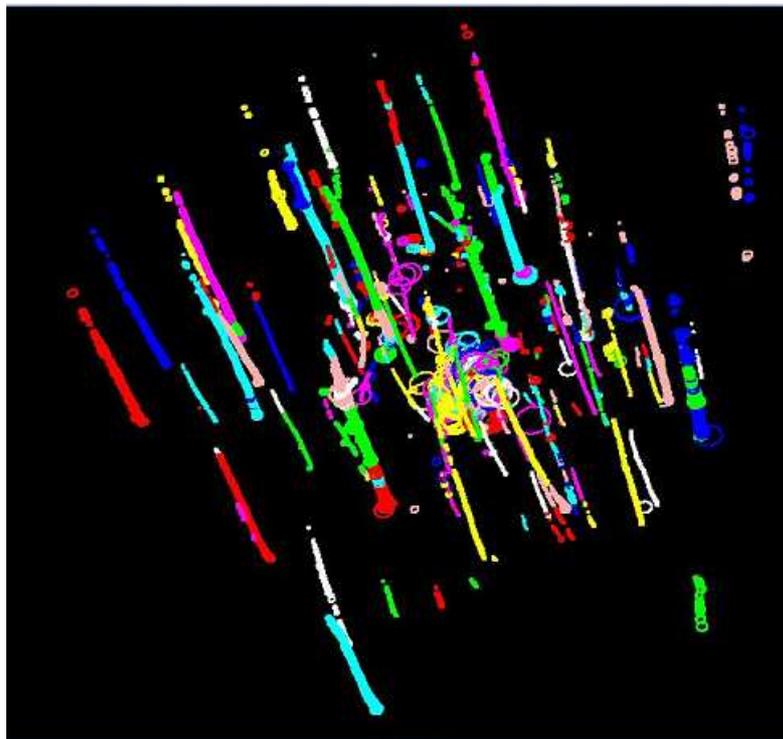


Figure 6: aperçu des arbres et des cercles obtenus par les algorithmes établis

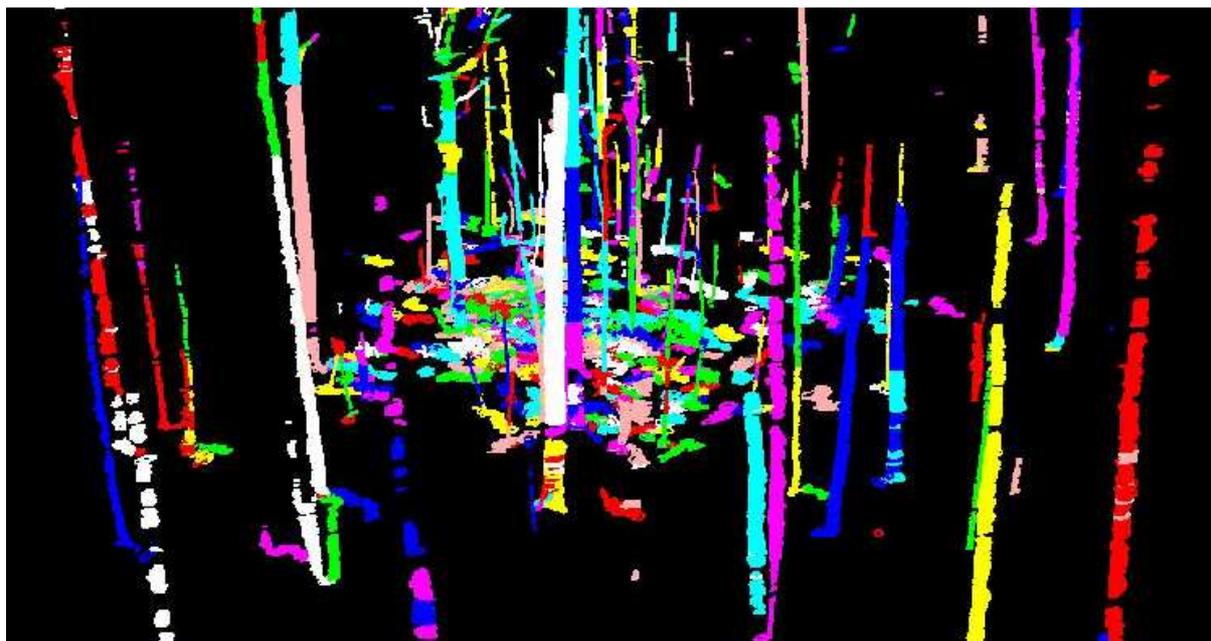


Figure 7: aperçu des ellipses et des arbres obtenus par les algorithmes établis

Observations :

- Le temps de traitement en couches du nuage complet est d'environ 43 minutes.
- Le gain temporel issu de la séparation du nuage en plage de distance est négligeable.
- Les arbres ne sont pas parfaitement reconstruits, on note qu'à une tige existante peut correspondre plusieurs arbres numériques. Une comparaison directe est actuellement compromise.
- Les arbres reconstruits à partir des cercles semblent mieux reconstruits que ceux qui ont été reconstruits par les ellipses.
- Le bruit semble mieux reconstruit par les formes elliptiques.
- Le nombre d'arbres numériques est supérieur au nombre d'arbres inventoriés.

Enfin, on pourra noter que l'export des cercles ne précise pas, à l'heure actuelle, à quel arbre numérique ils appartiennent de manière directe.

Interprétations :

Le temps de traitement est le premier, si ce n'est le plus important résultat de ce test : cette durée est beaucoup trop longue, le temps « bureau » actuel de traitement des données recueillies est inférieur à 10 minutes.

Le second enseignement de ce test est lié au fonctionnement actuel des étapes de traitement :

- L'export des résultats est à revoir pour réduire le temps de traitement. Dans l'état actuel des choses, il faut combiner les deux sorties séparées du traitement pour déterminer l'appartenance d'un cercle à un arbre numérique. Dès lors, on peut obtenir un DBH approché en calculant une moyenne sur des cercles reconstruits, cercles qui sont répartis sur une épaisseur supérieure au centimètre. Cela fait donc perdre tout l'intérêt de séparer le nuage en couches aussi fines (voir l'algorithme de traitement mis en place en annexe VI).

- La reconstitution des cercles est inférieure quantitativement à la reconstruction d'ellipses, mais semble au premier abord de meilleure qualité. En effet, l'ellipse présente deux degrés de liberté de reconstruction (les deux-demi grands axes), tandis que le cercle ne présente qu'une contrainte sur le rayon. Par conséquent, la reconstruction d'ellipses sera facilitée et moins précise que celle du cercle.
- La détection des tiges nous donne un plus grand nombre d'arbres reconstruits que d'arbres inventoriés : il s'agit donc d'être en mesure de les raccrocher.

La dernière remarque que l'on peut effectuer grâce à ces tests est liée à l'observation de la taille des données recueillies :

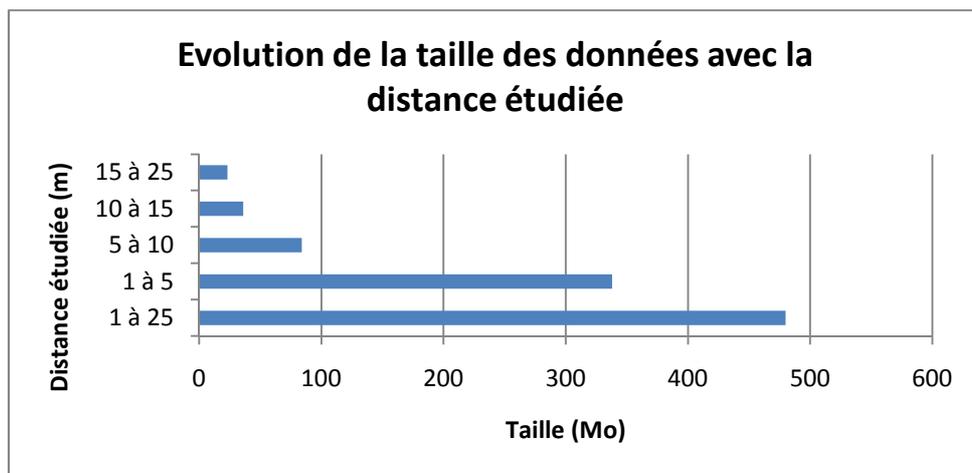


Figure 8: Evolution de la taille des données avec la fourchette de distance étudiée

On peut noter que 70,5% des données recueillies par la scanner appartiennent à la tranche 1-5 m, alors que dans cet inventaire, seules 4 des 37 tiges sont situées à une distance inférieure à 5 mètres. Cela signifie que les 33 tiges restantes se partagent les 29,5% de données restantes, non équitablement réparties : 14 arbres sont compris entre 5 et 10 m, 7 entre 10 et 15 m, 12 entre 15 et 25m.

On obtient le graphe suivant :

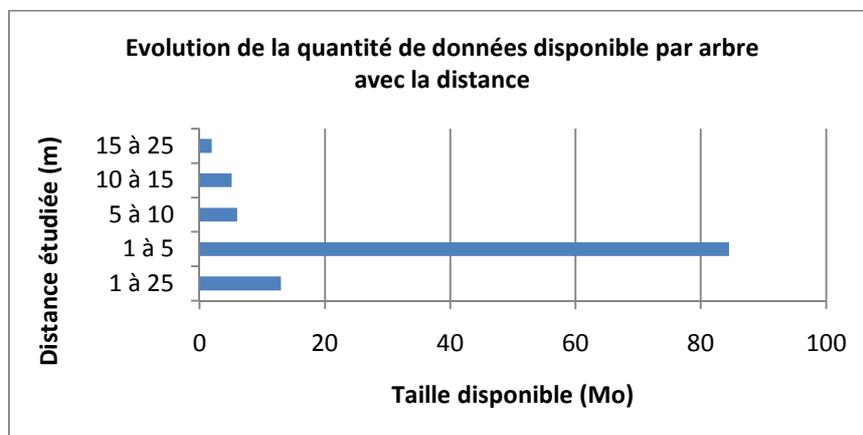


Figure 9: données disponibles par arbre

Les données étant très mal réparties, la clé sera donc de valoriser au mieux les données les plus lointaines.

4) Conclusion.

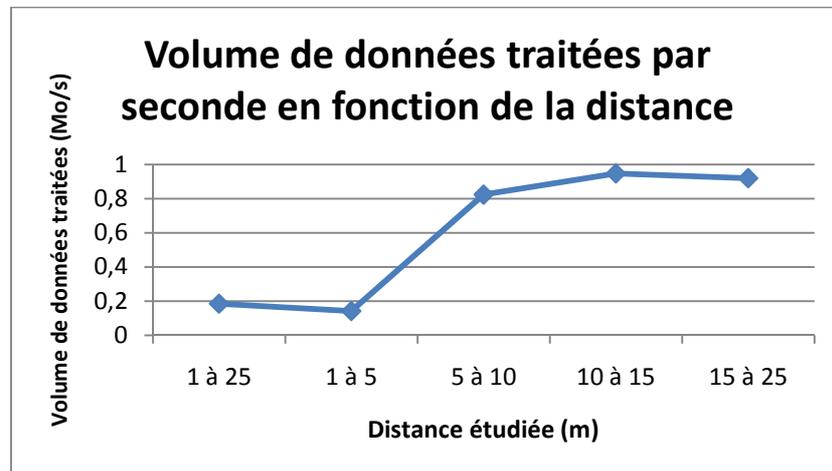


Figure 10: Vitesse de traitement des données

La version de COMPUTREE actuellement développée présente notoirement des problèmes de temps de traitement, liée à la vitesse du traitement.

Deux évolutions sont envisageables : changer de langage de développement, afin d'améliorer les performances ainsi que le temps de calcul ou travailler dans un environnement non plus 32 bits mais 64 bits, afin d'augmenter les quantités de mémoire disponibles quant au traitement des données.

L'objectif du projet étant une utilisation « universelle » de COMPUTREE, il a été décidé de travailler sur la première solution. C'est ainsi que le C++ dans la version Qt a été choisi, qui présente l'avantage d'être à la fois performant et transposable à tous les grands systèmes d'exploitation actuels (Windows, Linux, Unix, Mac).

Une dernière option aurait pu être envisagée : séparer le nuage pour rassembler ensuite les résultats. Il semble qu'il n'y ait pas de pertes d'information lors de la séparation (la somme des tailles des fichiers = taille du fichier complet), mais le fait de couper les données peut avoir un effet néfaste sur le traitement : en effet, une tige qui se trouve, à 1,30 m de hauteur, à une distance de 9,8 m, peut être, à 4 m de hauteur, situé à 10,2 m de distance. Cela fait que le bas sera traité dans un premier nuage et le haut dans un second. Le raccordement entre les deux arbres modélisés n'est pas évident et, enfin, le gain temporel est trop faible pour que la séparation du jeu de points s'avère nécessaire.

Dans un second temps, il est nécessaire d'améliorer l'export. Le double tableau de sortie (l'un directement exporté par la commande « Export », l'autre disponible dans la console de Eclipse) ne permet pas de faire un traitement rapide. L'export devra désormais être capable de distinguer les cercles reconstruits par arbre.

Il faudra ensuite travailler sur la reconstruction des arbres : en effet, il existe bien plus d'arbres numériques reconstruits que d'arbres inventoriés. Il faut ainsi trouver une méthode efficace pour améliorer la détection des tiges ou bien être capable de « recoller les morceaux » de manière efficace. Jusqu'à présent, les tiges sont détectées en partant des couches Z supérieures, il peut être donc envisagé de partir des couches inférieures ou médianes, ainsi que la mise en place d'une étape de groupement des arbres.

Enfin, les résultats obtenus par la régression elliptique nous a fait abandonner pour le moment l'utilisation de l'ellipse au profit du cercle. L'une des perspectives éventuelles serait de contraindre l'ellipse pour obtenir des résultats plus satisfaisants (demi-grand axe = 1,2 demi-petit-axe au maximum, etc.).

b) Développement sous C++ et travail avec des arbres témoin.

1) Vérification du gain apporté par l'utilisation du C++ lors de la conversion.

Afin de vérifier que la version codée en langage C++ est bien plus efficace que celle codée sous JAVA, il a été décidé d'effectuer le même test que précédemment sur la vitesse de traitement des données, avant de recoder l'intégralité de COMPUTREE en C++ :

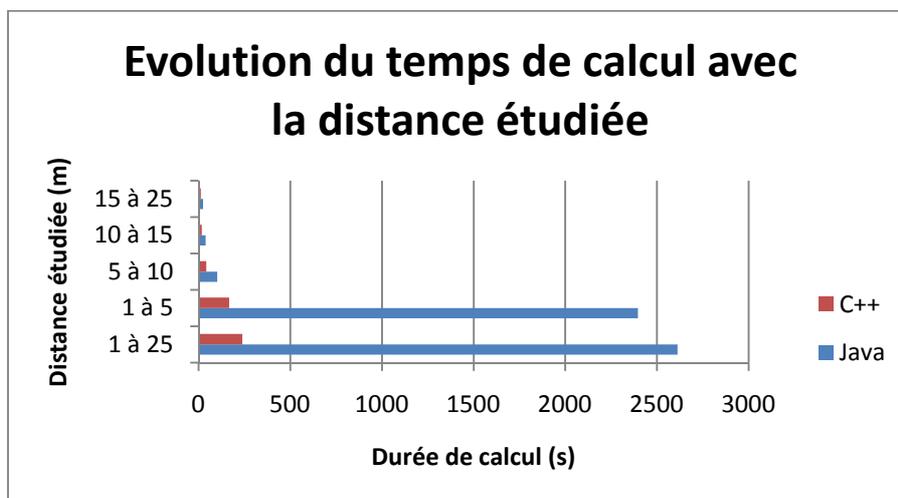


Figure 11: comparaison des temps de séparation du nuage témoin en couches.

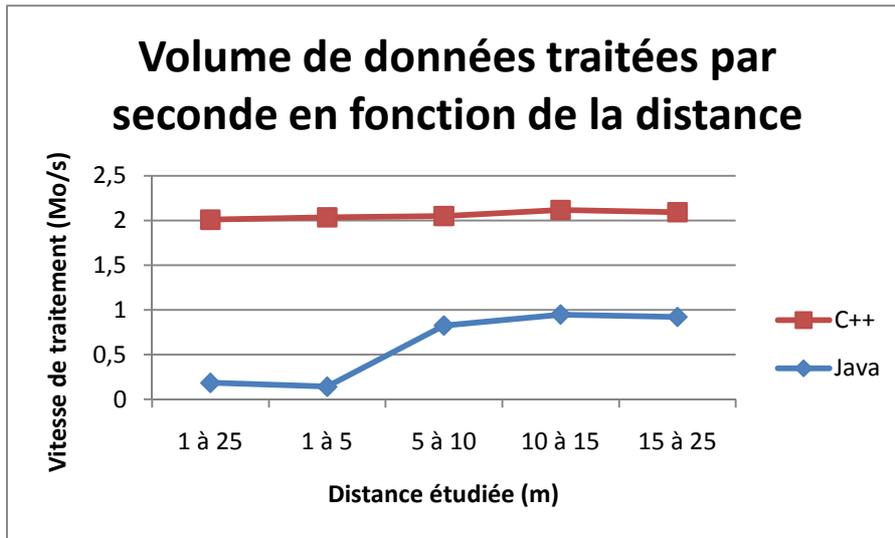


Figure 12: comparaison des vitesses de traitement des données

On note très aisément que le gain temporel est très important : on passe d'une conversion du nuage complet effectué en moins de 4 minutes, contre 41 minutes précédemment, ce qui nous donne une vitesse de traitement de 2 Mo/s contre 0,2Mo/s. Ainsi, au niveau de la conversion de données, cette version C++ est au moins 10 fois plus efficace que la version JAVA, son utilisation est donc bien nécessaire.

2) Codage en C++.

Le travail sous JAVA nous a permis de changer l'architecture de développement du projet. En effet, le développement sous JAVA tel qu'il a été codé, ne pouvait permettre la mise en place de nouveaux algorithmes sans un changement complet du code existant. Or dans une démarche de test/développement comme la nôtre, une plus grande flexibilité d'utilisation s'avère nécessaire. Il a donc été décidé de recoder complètement cette version du programme en C++ en conséquence. C'est ainsi qu'un système d'étapes a été mis au point : à une étape correspond un algorithme de traitement. Ainsi, l'enchaînement de ces étapes permet de choisir la chaîne de traitement que l'on effectue.

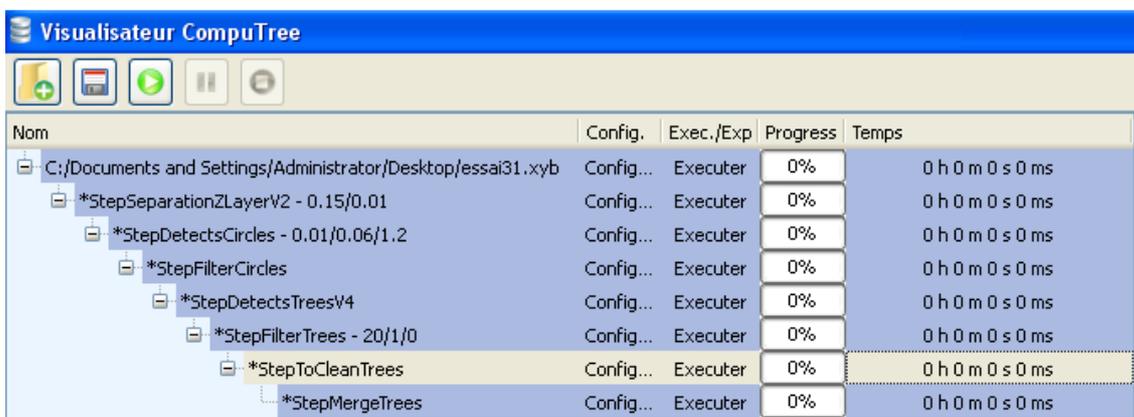


Figure 13: Choix des algorithmes de COMPUTREE codé en C++.

Les champs de la première colonne renseignent sur l'algorithme utilisé. Par exemple, l'étape « StepSeparationZLayerV2 » est une étape de séparation du nuage en couches d'épaisseur constante. Le V2 signifie quant à lui qu'il s'agit de la seconde version de l'algorithme. Enfin, la première version de cet algorithme existe toujours et reste un choix possible de traitement.

Ainsi, il est désormais beaucoup plus aisé d'ajouter des algorithmes et donc d'évaluer leur pertinence. Il en va de même pour les autres étapes du traitement.

Ce changement de langage nous a permis également d'effectuer plusieurs changements profonds dans l'algorithme :

- Désormais, le tri du jeu de points en couches selon l'axe Z se fait directement dans le logiciel, et ce juste après l'ouverture et la lecture du fichier.
- COMPUTREE ne nécessite donc plus de traitement externe, les fichiers binaires sortant directement des logiciels FARO sont utilisables.
- Il est également possible de changer l'épaisseur de la couche sur laquelle seront faits les groupements de points.
- La quantité minimale de points nécessaires pour faire un groupe (au départ paramétrable) est désormais fixée « en dur » à 10 points, ce n'est donc plus un paramètre.

Enfin, afin de confirmer l'intérêt du passage du langage JAVA au langage C++, les précédents jeux de tests ont été conservés et évalués à nouveau. Pour cette évaluation, les mêmes algorithmes ont été utilisés, et ce avec les mêmes paramètres.

En ne prenant que l'exemple du nuage complet, son temps de traitement est désormais de 90 secondes, la version nouvellement développée sous C++ est donc bien plus efficace que la précédente (qui présentait un temps de séparation en couches proche de 45 minutes).

3) Choix des arbres témoin.

Afin de pouvoir évaluer une « plage de bon fonctionnement des algorithmes », il a été choisi de travailler sur différents arbres témoins, devant représenter la majorité des cas qui peuvent intervenir sur ces placettes permanentes.

C'est ainsi qu'il a été choisi de travailler sur trois jeux différents : le premier contient un arbre de faible diamètre, le second un arbre de gros diamètre et le dernier un arbre double, les deux tiges ayant poussé sur la même souche. Les deux premiers jeux nous permettent d'évaluer les limites du paramétrage mis en place, le dernier les capacités de détection.

4) Plan d'évaluation et résultats.

Cette étude, qui porte sur trois types de tiges différentes, donne des résultats similaires sur les paramètres à employer. Ainsi, afin de ne pas répéter trois fois consécutivement la même démarche dans ce rapport, nous n'allons traiter dans la suite que le cas de la petite tige.

L'estimation des paramètres s'est faite de la manière suivante :

- On détermine la meilleure (ou les meilleures) valeur(s) pour le premier paramètre.
- On fixe le second paramètre en utilisant les résultats de l'évaluation précédente.
- On continue la même démarche jusqu'au dernier paramètre.

Les paramètres ont ainsi été évalués dans l'ordre suivant : « Distance maximale d'un point à un groupe », « Epaisseur des couches en Z », « Erreur maximale en rayon », « Rayon minimal à observer », « Rayon maximal à observer », « Nombre de cercles minimum pour faire un arbre ».

La qualité du résultat s'est portée sur la valeur estimée du DBH obtenu par les algorithmes par rapport au DBH inventorié (ce qui permet d'évaluer la précision du résultat), sur le nombre de cercles reconstruits et sur le nombre d'arbres numériques reconstruits (ce qui permet d'estimer la qualité de reconstruction des algorithmes).

L'intégralité des résultats est disponible en annexe VII.

L'inventaire effectué nous donne un arbre de circonférence égale à 437 mm, soit un diamètre de 139,1 mm.

Les essais 1 à 9 nous ont permis de fixer la distance maximale pour qu'un point soit inclus dans un groupe. Compte-tenu des résultats obtenus, notre choix aurait pu se porter sur le test 8, qui présente le diamètre reconstruit le plus proche et un nombre de cercles suffisants. Cependant, l'objectif n'étant pas d'avoir à fixer un paramètre pour une tige (on peut remarquer que la distance de 12 cm correspond au diamètre recalculé de l'arbre), nous nous sommes arrêtés sur les valeurs de ce paramètre bien moins liées au diamètre mesuré. Ainsi, nous nous sommes orientés sur les résultats obtenus sur les tests 1, 2 et 7, à savoir les distances de 15, 30 et 50 centimètres.

Les essais 10 à 15 nous ont permis de tester la variation d'épaisseur de la couche en Z. On peut noter qu'une épaisseur de 5 cm de nous ne fournit pas de résultat (pas de diamètre reconstruit), tandis que l'épaisseur de 2 cm nous donne des résultats plus précis, mais en plus faible quantité (60 cercles reconstruits pour 1 cm d'épaisseur, et 25 pour 2 cm, le produit cercle X épaisseur est donc plus faible dans le second cas). Nous avons donc conservé l'épaisseur de 1 cm pour nos calculs.

Les tests 16 à 27 évaluent la pertinence du paramètre concernant l'erreur maximale de reconstruction autorisée pour les cercles. Une erreur trop faible ne reconstruisant aucun cercle, il s'agit de trouver une valeur qui nous permet de garder suffisamment de précision sans augmenter trop fortement le bruit. On note qu'en choisissant 0,01, 0,05 et 0,1 comme erreur autorisée, la quantité de cercle reste constante, mais que le diamètre calculé et le bruit augmentent avec l'erreur. L'erreur la plus faible a donc été conservée (0,01).

Les tests 28 à 33 évaluent la pertinence du paramètre du rayon minimal reconstruit. Lorsque l'on prend une valeur faible, on reconstruit très bien les tiges, mais pour un arbre détecté, plusieurs arbres numériques correspondent. En prenant une valeur trop grande, les résultats obtenus se sont révélés incohérents (les diamètres obtenus valent plus du double du diamètre relevé). On remarque également que la valeur de 15mm pour la distance maximale pour qu'un point appartienne à un groupe présente des problèmes lors de ce test de rayon minimal de reconstruction. Lorsqu'on augmente le rayon, rien n'est reconstruit, tandis que lorsque l'on diminue le rayon, les résultats sont

incohérents. Ainsi, dans la suite de cette évaluation, nous avons décidé de nous passer de cette valeur, pour nous concentrer sur des valeurs de distance de 30 et 50 cm.

Les tests 34 à 37 évaluent la pertinence du rayon maximal reconstruit. On note que pour cette petite tige, le fait d'augmenter le rayon maximal ne change rien à la valeur reconstruite. On aura ainsi tendance à prendre le rayon maximal, afin de pouvoir mieux évaluer les plus grosses tiges.

Enfin, les tests 38 à 41 évaluent la pertinence du nombre de cercles minimum pour reconstruire un arbre. Les résultats sont constants, on va donc avoir tendance à utiliser la valeur intermédiaire, soit 20.

En définitive, on peut conclure que le meilleur paramétrage pour notre cas est :

50 cm pour la distance maximale d'un point à un groupe, une épaisseur de couche de 1 cm, une erreur maximale de reconstruction de 0,01 m, un rayon minimal reconstruit de 2 cm, un rayon maximal reconstruit de 100 cm, un nombre minimal de cercles de 20.

c) Evaluation sur des jeux de données complets.

Compte-tenu des résultats prometteurs aperçus lors de la détermination des paramètres, il fallait nécessairement effectuer des essais sur des placettes complètes et inventoriées. Pour cela, nous avons travaillé avec les valeurs des paramètres obtenues précédemment.

1) Choix des placettes.

7 placettes ont été choisies parmi la base de données disponible à l'ONF de Velaine-en-Haye. Elles présentent les caractéristiques suivantes :

- Une densité de peuplement apparente variée.
- Des peuplements variés.
- Des diamètres variables selon les distances.
- Plusieurs essences par placette.
- Des densités intra-placettes variables.

2) Algorithmes mis en œuvre.

Il a été nécessaire de mettre en œuvre, en parallèle du développement du programme, des algorithmes qui peuvent utiliser les données qui sont issues de COMPUTREE. Ils ont ainsi été développés sous R (algorithmes disponible en annexe VIII).

1) La comparaison visuelle.

Présentant une interface graphique directement sur son environnement de travail ainsi qu'un langage aisé d'apprentissage, R permet de mettre en place et de tester très rapidement les algorithmes complémentaires de traitement des données.

La comparaison visuelle permet d'estimer dans un premier temps l'écart d'angle et de position existant entre les jeux. En effet, l'inventaire a été effectué à partir du nord magnétique, et l'angle 0 du scan ne lui correspond pas. Un recalage angulaire doit donc être effectué, de même qu'un recalage de positionnement, compte-tenu du fait que le centre de la placette n'est pas forcément le point de prise de mesure du scanner laser. Ces deux modifications sont effectuées manuellement.

L'interface graphique permet également d'estimer de façon qualitative les données issues de COMPUTREE, et donc de les comparer aux données d'inventaire.

2) La comparaison de localisation.

Pour déterminer la précision de localisation, il faut être en mesure, dans un premier temps, de localiser puis de comparer le lieu des arbres.

La localisation varie selon le jeu de données à traiter : l'inventaire est localisé en azimuth-distance, il faut donc convertir ces données en coordonnées cartésiennes pour, d'une part, les tracer sur la fenêtre de résultat de R et d'autre part les comparer aux données issues du logiciel.

Une fois les coordonnées transformées et le nuage recalé, il s'agit de comparer la localisation des arbres inventoriés par rapport à celle des arbres reconstruits et inversement. Des essais préliminaires montraient qu'il existait sur ces placettes plus d'arbres détectés qu'inventoriés. Nous avons alors appliqué le principe suivant:

- Pour chaque arbre inventorié, il faut déterminer quel est l'arbre numérique le plus proche.
- Pour chaque arbre détecté, il faut déterminer quel est l'arbre inventorié le plus proche.
- S'il existe un couple d'arbres qui présentent ces caractéristiques, alors on considère que l'arbre numérique représente cet arbre inventorié et que ce dernier est bien reconstruit par les algorithmes.

Cette considération de distance n'est valable que si les valeurs sont inférieures à 1,5 mètres. En effet, il est possible que des erreurs de mesure aient « délocalisé » les arbres inventoriés, et que cette erreur soit proche du mètre cinquante (en combinant les erreurs de reconstruction, de mesure, de recalage et de détection des cercles). Au-delà, on considère que l'arbre inventorié n'a pas été détecté.

Dans les cas où à un arbre détecté correspondait plusieurs arbres inventoriés, on a cherché à évaluer la tige qui ressemblait le plus à l'arbre détecté (évaluation en diamètres), on éliminait ainsi les arbres problématiques.

3) La comparaison des diamètres.

Dans la version utilisée pour cette évaluation, il n'existait pas de possibilité de sortie de tranches, mais uniquement des cercles obtenus par COMPUTREE. Il a donc fallu mettre en œuvre un algorithme permettant, non seulement de retrouver les arbres parmi l'ensemble des cercles obtenus, mais aussi de déterminer un diamètre à 1,30m de hauteur à partir du cercle le plus bas reconstitué.

Les cercles n'étant pas nécessairement présents à 1,30 m de hauteur, il a été choisi d'effectuer une interpolation à cette hauteur pour reconstruire les DBH, interpolation effectuée entre les deux cercles encadrant le 1,30 m.

4) La comparaison de reconstruction.

Comme évoquée précédemment, l'un des premiers problèmes observés a été le fait qu'à un arbre réel puisse correspondre plusieurs arbres numériques. Il a fallu donc mettre en œuvre, sur le logiciel COMPUTREE, un algorithme de raccrochage des arbres.

La première piste évoquée était de prendre, autour de chaque arbre « source » détecté, un carré d'un mètre de côté, et de considérer que tout ce qui appartenait à ce carré faisait partie du même arbre. Sachant que la majorité des arbres numériques qui ne correspondent pas à un arbre réel font partie du houppier, il a été choisi comme critère d'arbre source l'arbre qui avait le cercle reconstruit le plus bas.

Cependant, cette démarche n'est pas sans poser un problème majeur : lorsque l'on travaille sur un arbre double ou sur une tige présentant une fourche à une faible hauteur, l'algorithme raccrochera l'ensemble de ce qui a été détecté, ce qui ne correspond bien entendu pas du tout à la réalité du terrain.

Il a donc été choisi une autre approche, l'approche vectorielle. Pour chaque arbre numérique, on détermine le vecteur qui relie les centres du cercle le plus bas et du cercle le plus haut. On évalue dès lors la distance minimale entre les deux vecteurs, et si celle-ci est inférieure à un palier défini, on raccroche les deux tiges.

Par ailleurs, il est également possible que, malgré cette étape de raccrochage, les arbres ne soient pas parfaitement ajoutés. On a donc également cherché à évaluer le nombre d'arbres numériques qui correspondent à un arbre inventorié.

3) Résultats.

Tableau 3: Résultats des tests effectués sur 7 placettes

Nom de la placette	Arbres non reconstruits	Arbres inventoriés	Arbres détectés	Moyenne des distances entre un arbre inventorié et l'arbre reconstruit le plus proche (m)	Distance maximale inventoriée (m)	Nombre d'arbres qui peuvent correspondre à une tige inventoriée	Reste
6p2d	0	29	172	0,51	20,94	70	102
6p2g	1	43	180	0,33	20,4	113	67
6p7g	0	22	190	0,70	19,49	59	131
12p2g	0	35	238	0,20	20,67	104	134
13p6g	1	42	314	0,40	19,34	142	172
13p10g	1	31	227	0,39	19,79	107	120
16p4d	0	37	251	0,35	20,78	110	141
Total	3	239	1572			705	867

On appelle arbre non reconstruit une tige inventoriée qui ne comporte pas d'arbres numériques à moins de 1,5 m de distance.

Les arbres inventoriés proviennent des inventaires réalisés et les arbres détectés sont le résultat du traitement réalisé par COMPUTREE.

La moyenne des distances nous permet d'estimer la précision de la reconstruction sur l'ensemble des tiges détectées, et ce après le recalage angulaire effectué. La distance maximale inventoriée nous donne le diamètre d'étude de la placette.

Le nombre d'arbres qui peuvent correspondre à une tige inventoriée correspond au nombre d'arbres numériques qui sont présents dans un cercle d'1,5 m autour de la tige inventoriée, et le reste correspond à des arbres détectés dont on ne peut rien dire (non inventoriés, bruit ?), par manque de données.

On peut effectuer différentes interprétations de ces résultats :

- Le nombre d'arbres inventoriés à qui ne correspond aucun arbre détecté est très faible, les algorithmes semblent ainsi avoir un très fort pouvoir de détection.
- Le nombre d'arbres détectés est très supérieur au nombre d'arbres inventoriés (1572 contre 239).
- La moyenne des distances de reconstruction est inférieure au mètre, la localisation est donc efficace.
- 705 arbres numériques peuvent correspondre aux arbres inventoriés (239).

4) Conclusion.

On peut tirer du tableau précédent plusieurs conclusions :

- S'il existe 705 arbres numériques qui peuvent correspondre à 239 arbres inventoriés, c'est que le raccrochage des tiges n'est pas suffisamment efficace à l'heure actuelle. Il s'agira de l'améliorer.
- 867 arbres numériques sont éloignés d'une distance supérieure à 1,5 m de tous les arbres inventoriés. Il est par ailleurs aisément visible sur la fenêtre de visualisation de COMPUTREE que certains arbres ont été pointés par le TLS alors qu'ils n'ont pas été inventoriés. On peut ainsi penser que les données dont nous disposons sont actuellement non exhaustives, il nous faut donc disposer d'un jeu de données le plus précis possible pour effectuer la meilleure évaluation possible de nos outils.
- Il y a très peu d'arbres qui ne sont pas détectés, les algorithmes jusqu'à présents utilisés sont donc performants, il n'est pas nécessaire de tout refaire mais de peaufiner les résultats actuels.

L'objectif suivant est donc de mettre en place un jeu de données satisfaisant, puis de lui appliquer les algorithmes actuellement utilisés pour pouvoir juger de leur efficacité réelle. C'est ainsi que nous avons choisi de mener notre campagne d'essais en forêt de Bezange.

IV. Résultats et discussions.

1) Version « finale » de COMPUTREE.

Compte-tenu des différentes remarques formulées précédemment ainsi que des réflexions menées au fur et à mesure du projet, de nombreuses modifications ont été apportées au programme : la version « finale » du logiciel (la version existante à l'heure actuelle) présente ainsi un certain nombre d'étapes décrites dans la suite de cette partie. Les copies d'écran décrivant ces étapes sont disponibles en annexe IX.

Lecture du fichier de points.

La lecture du fichier de points permet de visualiser l'ensemble des points retrouvés par le scanner.

Séparation en couches et de détection de formes.

Ces deux algorithmes sont désormais regroupés sous une seule étape. Deux choix sont possibles : la méthode originelle de calcul (distance d'un point au groupe, épaisseur de la couche à choisir) ou la nouvelle qui, en plus des paramètres que l'on vient d'évoquer, nous permet de limiter la distance des points par rapport au centre du groupe.

Détection des cercles.

Cette étape n'a pas évolué depuis le début, et ce malgré avoir abordé d'autres pistes. En effet, une recherche bibliographique menée en ce sens a permis de mettre en évidence l'algorithme de Delogne-Kasa [5], qui utilise une estimation par maximum de vraisemblance pour reconstituer des cercles dans des parties de nuage. Les deux algorithmes ont alors été testés sur différents jeux de données (à savoir des tranches de 1cm), et le gain résultat du passage à l'estimation par maximum de

vraisemblance est de 0,5mm en moyenne, et ce sur des diamètres variables (allant de 20 cm à 60 cm). Il a donc été choisi de conserver la résolution en termes des moindres carrés, l'algorithme proposé de donnant pas un avantage significatif par rapport à l'autre.

Pré-nettoyage des cercles.

Cette étape permet de limiter l'impact des cercles aberrants, c'est-à-dire ceux qui possèdent un diamètre beaucoup trop faible ou beaucoup trop élevé et qui ne correspondent finalement à rien.

Détection des arbres.

Jusqu'à présent, les arbres étaient détectés en partant du haut du nuage. Afin de vérifier la validité de cette hypothèse, il a été choisi de travailler à partir des couches inférieures ou médianes. Après plusieurs essais menés sur les jeux de données, il semble que la détection la plus efficace soit celle qui passe par le haut des tiges. Il a donc été choisi de conserver cette méthode de détection. Les algorithmes restent cependant disponibles.

Filtrage des arbres.

Il est toujours possible de filtrer les arbres numériques reconstruits en fonction de leur nombre de cercles.

Nettoyage des arbres.

Jusqu'à présent, il n'y avait aucune contrainte sur les diamètres reconstruits. Cependant, au fur et à mesure de l'évolution du projet, nous avons observé que des cercles aberrants apparaissaient sur les arbres reconstruits, et ce qu'il soit masqué ou non. Ceci est dû à un effet de bord généré par les branches et les feuilles, qui ont tendance à faire grandir fortement le cercle reconstruit. Il a donc été décidé d'implémenter une étape de filtrage des diamètres. On calcule la moyenne des diamètres reconstruits sur un même arbre, et si le cercle considéré a un diamètre supérieur ou inférieur à un pourcentage défini de la moyenne, on le supprime de l'arbre.

Raccrochage des arbres.

Cette étape n'a pas évolué, elle met toujours en œuvre l'évaluation de distance entre les vecteurs directeurs des arbres reconstruits.

Reconstitution de tranches.

Le programme était, jusqu'à présent, capable uniquement de ressortir les cercles reconstruits. Afin d'éviter un traitement supplémentaire sur R, l'obtention de tranches a été implémentée dans COMPUTREE, selon la même méthode d'interpolation linéaire que précédemment. Cette étape de tranchage de jeux de données nous donne le choix entre :

- Une tranche contenant les cercles reconstruits à un Z constant du nuage
- Une tranche contenant les cercles reconstruits à une hauteur H du point le plus bas de chaque arbre reconstruit.

2) Résultats de la campagne de validation.

a) Inventaire.

Chaque placette a pris une journée pour être inventoriée (7 heures pour la placette 221, 5h30 pour la 222). Compte-tenu de la précision d'inventaire que l'on recherchait, nous avons relevé toutes les tiges dont le diamètre était supérieur à 3 cm pour la placette 221 et à 5 cm pour la placette 222.

La placette 221 présente 85 arbres inventoriés et 11 essences différentes.

La placette 222 présente 69 arbres inventoriés et 7 essences différentes.

Le travail sur la prise de mesure sur les arcs de cercle n'a pas été effectué complètement : en effet, l'acquisition de cette donnée allongeait beaucoup trop le temps de mesure, ainsi nous n'avons pas évalué les capacités d'obtention des points du scanner laser terrestre.

b) Traitement des données.

L'enchaînement d'étapes mis en place pour effectuer le traitement des données issues de ces placettes est le suivant :

- On sépare le nuage en couches de 1 cm d'épaisseur, la distance maximale d'un point à un groupe est de 15 cm et la distance maximale d'un point au barycentre du groupe est de 50 cm.
- La détection de cercles admet une erreur maximale de 0,01 cm, un rayon minimum détecté de 2 cm et un rayon maximum de 120cm.
- Les arbres possédant moins de 50 cercles sont supprimés.
- Le nettoyage des arbres se fait sur les cercles ayant une différence de plus ou moins 20% avec le diamètre moyen de la tige.
- Les arbres sont raccrochés selon la méthode vectorielle précédemment citée.
- L'étude sur une tranche du nuage reconstruit se fait à $Z = 0$.

L'étape de traitement sous le logiciel R a quant à elle évoluée, principalement sur le choix des données prises en compte pour l'analyse. En effet, les essais menés sur les 7 jeux de données précédents ont montré qu'il existait un problème de raccrochage des arbres numériques. C'est la raison pour laquelle un nouvel algorithme a été implémenté, qui permet d'éliminer les tiges reconstituées qui sont ont un Z minimum situé au-delà de 2,5 mètres de hauteur :

- On lit, en plus du fichier d'inventaire et du fichier de cercles reconstruits, le fichier contenant l'intégralité des cercles reconstruits, labélisés par arbre numérique.
- On calcule pour chacun de ces arbres sa hauteur minimale (elle correspond au cercle le plus bas reconstruit).
- Si cette hauteur minimale est inférieure à 2,5 m, l'arbre est utilisé et le cercle issu de COMPUTREE est conservé. Si ce n'est pas le cas, nous n'en tenons pas compte.

Dès lors, il ne nous reste que les arbres inventoriés ainsi que les arbres détectés filtrés (selon ce dernier algorithme), qui sont comparés via la méthode proposée précédemment, et ce après un recalage angulaire.

Il reste, malgré le traitement mis en place, quelques coquilles sur nos données, ces dernières ont alors été évaluées une par une de façon manuelle, de manière à obtenir un inventaire détecté le plus proche possible de l'inventaire réel (voir Annexe X).

c) Résultats.

1) Sur la détection.

Tableau 4: Résultats par placette de la détection et de la localisation des arbres

Placette	Arbres inventoriés	Arbres détectés	Arbres utilisables	Arbres localisés auto	Arbres bien localisés auto	Arbres localisés	Inventaire non localisé	Détection non localisée
221	85	147	95	59	51	59	26	37
222	69	127	83	52	47	57	12	26

Ce tableau nous permet d'observer rapidement le résultat des algorithmes mis en œuvre.

« Arbres inventoriés » correspond au nombre d'arbres relevés sur les placettes.

« Arbres détectés » correspond au nombre d'arbres détectés par COMPUTREE.

« Arbres utilisables » correspond au nombre d'arbres utilisables après le filtrage selon la hauteur minimale de l'arbre reconstruit.

« Arbres localisés auto » correspond au nombre d'arbres recalés par l'algorithme de localisation.

« Arbres bien localisés auto » correspond au nombre d'arbres réellement bien localisés.

« Arbres localisés » correspond au nombre total d'arbres localisés (après correction manuelle).

« Inventaire non localisé » correspond au nombre d'arbres inventoriés non reconstruits.

« Détection non localisée » correspond au nombre d'arbres détectés non localisés.

On note, tout d'abord, que les algorithmes qui ont été mis en place depuis les derniers tests sont plus efficaces, le ratio détection/inventaire s'est considérablement réduit (ce résultat est dû majoritairement à un filtrage des arbres plus important ainsi qu'au raccrochage des tiges).

Par ailleurs, l'étape de localisation est assez efficace, à savoir que 88% des arbres considérés comme bien localisés le sont effectivement. Cette étape reste cependant à être améliorée : les pistes envisagées sont présentées à la fin de ce rapport, dans la section « Perspectives ».

Les cartes de détection sont disponibles en annexe XI.

2) Etude de la non-détection.

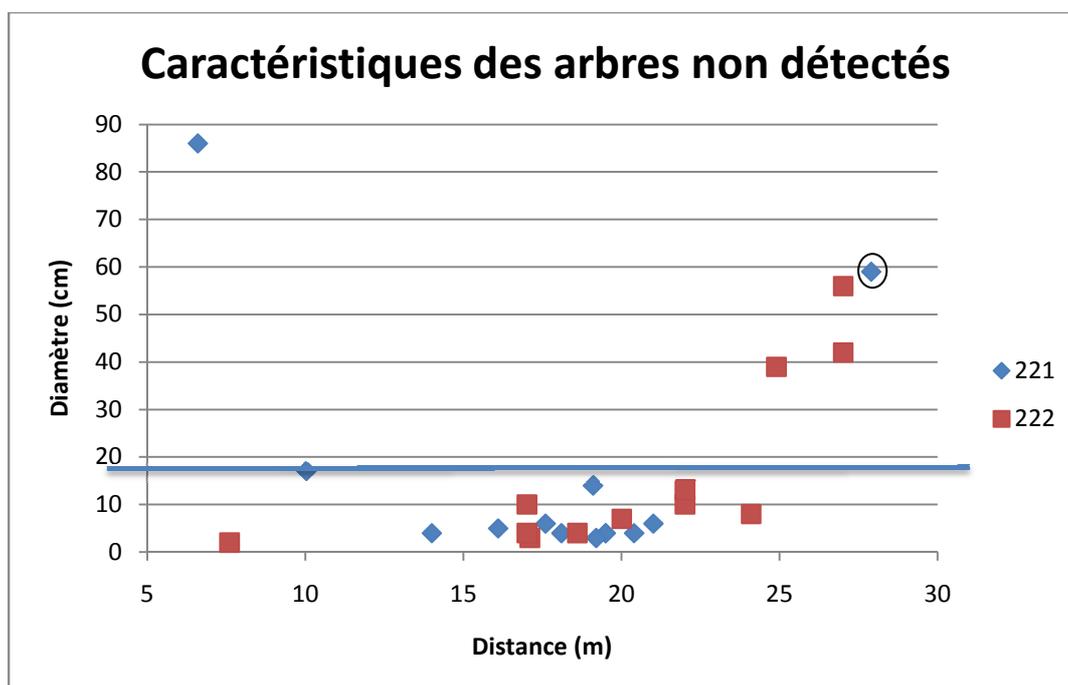


Figure 14: Caractérisation des arbres non détectés.

Ce graphique nous donne les distances et les diamètres des tiges inventoriées non reconstruites. On peut noter que 5 des tiges inventoriées au-delà du diamètre de pré-comptage ne sont pas reconstruites. Sur 4 d'entre elles, on voit aisément que chacune est invisible à l'œil nu, le scanner terrestre ne doit obtenir sur ces arbres que peu de points, ce qui fait que la reconstruction est quasiment impossible :



Figure 15: Exemple d'arbre non détecté

La figure 14 nous montre la vision que le scanner a de la tige qui est entourée sur la figure 13. Le rectangle nous donne la position de cet arbre, et on voit très nettement qu'il est impossible de distinguer un arbre parmi les très petites tiges, les branches ou encore les feuilles présentes sur la scène.

Le seul réel problème vient du très gros bois (diamètre supérieur à 67,5 cm) de la placette 221. Comme on peut l'observer sur la figure 27, il existe, autour de cette tige, des arbres reconstruits dont le diamètre est inférieur au diamètre de pré-comptage (17,5 cm). Or, lors de la phase de rattachement des arbres, ces petites tiges se rattachent au gros arbre : il est donc bien reconstruit, mais le cercle détecté à une hauteur $Z = 0$ correspond à ces petites tiges. On obtient ainsi le résultat suivant :

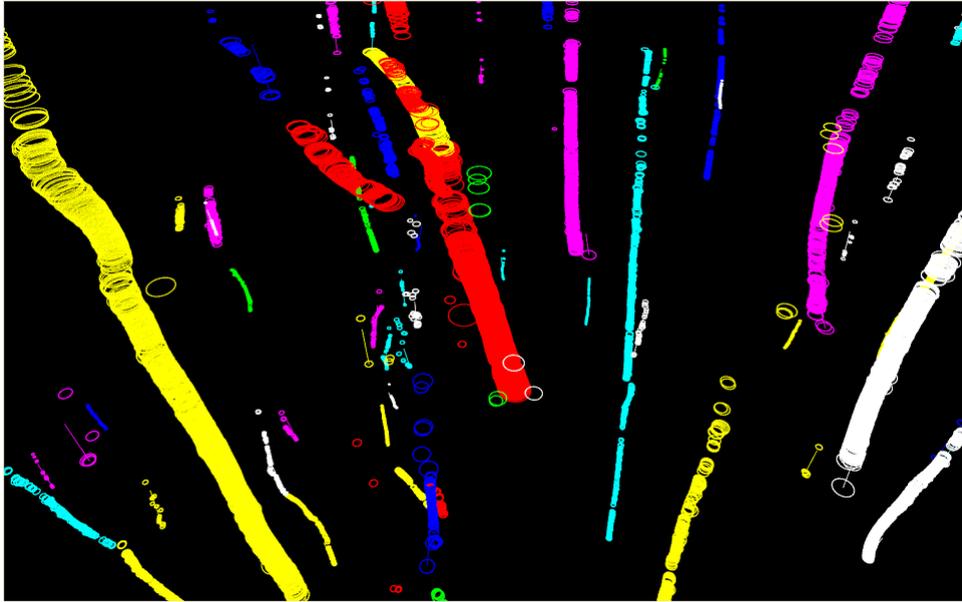


Figure 16: Détection des arbres de la placette 221 (le très gros bois problématique est de couleur rouge)

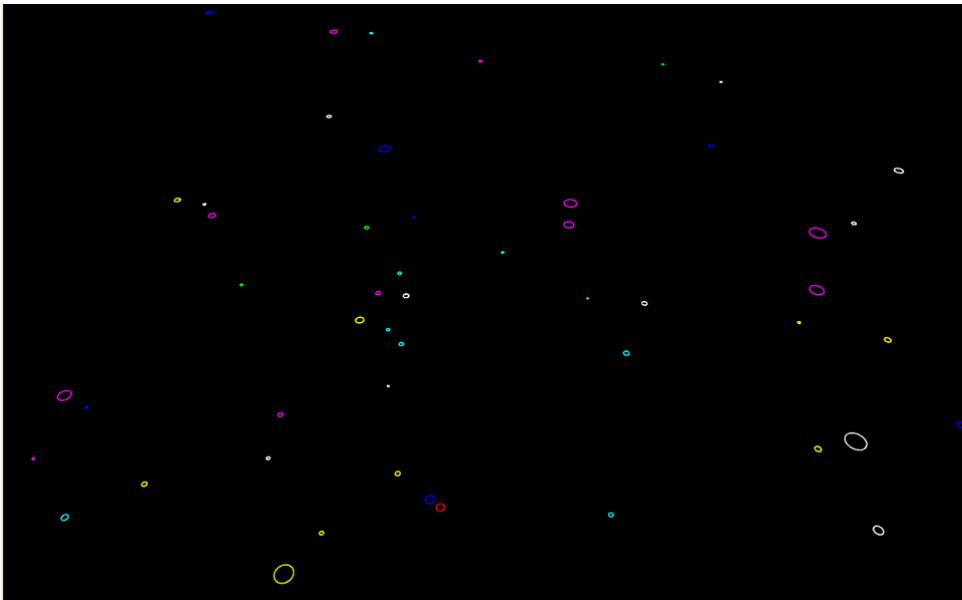


Figure 17: Cercles à $Z=0$ détectés pour la placette 221 (le cercle du très gros bois est en rouge)

Enfin, le graphique de la figure 13 nous montre que la majorité des tiges non détectées sont situées sous le diamètre de pré-comptage. Il existe deux explications à ce phénomène :

- Pour les tiges les plus éloignées, la non-détection est due aux ombrages divers, d'où un manque de points.

- Pour les arbres les plus proches, la non-détection est issue de problèmes de raccrochage d'arbres numériques. En effet, ces arbres de faible diamètre et inventoriés font partie de cépées très proches de tiges de gros diamètre. L'algorithme de raccrochage les agglomère à ces arbres, ce qui fait qu'ils ne sont pas détectés.

Compte-tenu des résultats obtenus, on peut considérer que la non-détection est assez faible. Les non-reconstructions proviennent majoritairement d'un manque de points sur les arbres observés ou d'un problème lié au raccrochage des tiges détectées.

3) Etude de la détection en classes de diamètre.

Il faut désormais étudier la capacité des algorithmes pour retrouver les diamètres inventoriés. Ainsi, nous allons étudier, dans un premier temps, l'obtention des classes de diamètre.

Tableau 5: Inventaire et détection en classes de diamètre, placettes 221 et 222.

	Placettes 221 + 222					Total
	D < 17,5	17,5 < D < 27,5	27,5 < D < 47,5	47,5 < D < 67,5	D > 67,5	
Inventaire manuel	76	30	28	16	4	154
Inventaire détecté	48	30	26	14	3	121
Arbres reconstruits	51	35	26	9	0	121

« Inventaire manuel » correspond aux arbres inventoriés sur le terrain.

« Inventaire détecté » correspond aux arbres inventoriés sur le terrain qui ont été détectés.

« Arbres reconstruit » correspond aux classes de diamètre reconstruites par l'algorithme.

Sur ces deux placettes, on note qu'une grande partie des arbres inventoriés sont des tiges qui ne sont généralement jamais évaluées (à savoir le sous-étage et les tiges de parcelles dites en régénération).

En ce qui concerne les tiges détectées, on voit très nettement que la plus grande perte de données s'effectue sur ces arbres de diamètre inférieur à 17,5 cm, ce qui, pour la sylviculture actuelle, n'est pas une perte de données vitales. Par contre, pour les tiges de plus gros diamètre (au-dessus du diamètre de pré-comptage), les résultats sont encourageants : pour les 78 arbres inventoriés de ces classes, 73 d'entre eux ont été détectés. Enfin, on a recréé 69 arbres dont les diamètres reconstruits sont supérieurs à 17,5 cm. Cela montre que l'on peut reconstruire un certain nombre de diamètres, cependant rien n'indique, dans cette étude sur la localisation, que ces diamètres détectés correspondent bien aux diamètres inventoriés correspondants.

Une nouvelle analyse des données obtenues nous donne les résultats suivants :

Tableau 6: Erreur de reconstruction des diamètres selon la classe

	D < 17,5	17,5 < D < 27,5	27,5 < D < 47,5	47,5 < D < 67,5	D > 67,5
Nombre de tiges qui ne correspondent pas à la classe	7	10	7	5	1
Perte/gain en tiges de la classe	3	2	-1	-3	-1

La première ligne de ce tableau permet d'évaluer le nombre de tiges détectées qui ont un diamètre détecté qui ne correspond pas au diamètre inventorié, tandis que la seconde nous donne l'écart entre le nombre de tiges qui entrent dans la classe et le nombre de celles qui en sortent. On peut noter que les classes de faible diamètre sont privilégiées : il semble que nos algorithmes sous-dimensionnent les diamètres.

La confirmation de cette observation nous est apportée par ce tableau :

Tableau 7: "Echange" de tiges entre les classes de diamètre

	D < 17,5 devient	17,5 < D < 27,5 devient	27,5 < D < 47,5 devient	47,5 < D < 67,5 devient	D > 67,5 devient
D < 17,5	0	4	1	0	0
17,5 < D < 27,5	2	0	3	1	0
27,5 < D < 47,5	0	0	0	3	0
47,5 < D < 67,5	0	0	0	0	1
D > 67,5	0	0	0	0	0

Il nous présente les transformations que subissent les tiges inventoriées : la partie au-dessus de la diagonale du tableau représente la perte en diamètre des arbres inventoriés, tandis que le triangle inférieur nous donne le nombre de tiges qui gagnent en diamètre. Le sous-dimensionnement est ici bien observable (13 ont une perte de diamètre contre 2).

4) Etude de la détection des diamètres.

Nous étudions désormais la détection des diamètres : le graphique suivant nous donne l'évolution des diamètres inventoriés en fonction des diamètres détectés.

Diamètres inventoriés en fonction des diamètres détectés (cm)

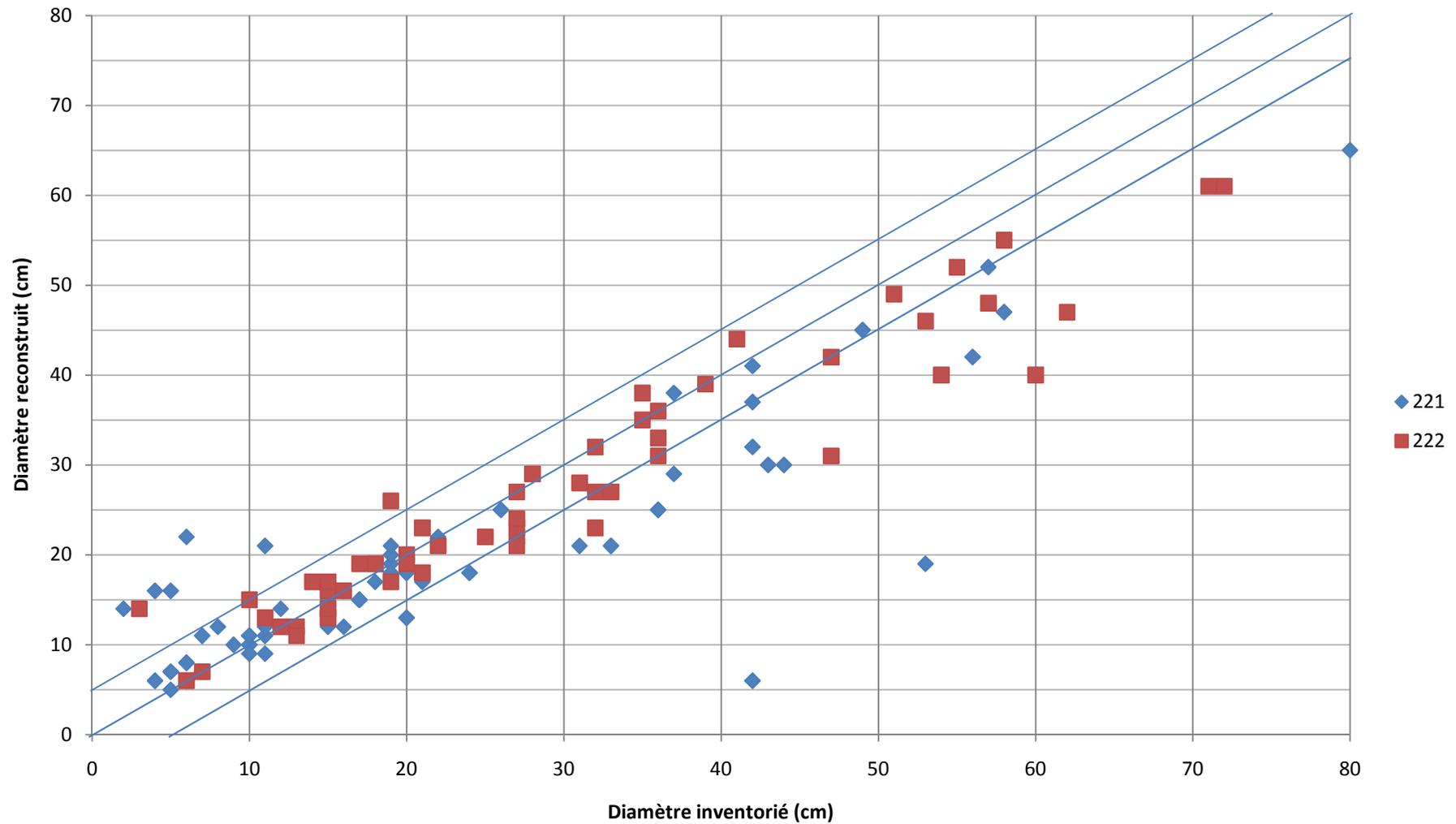


Figure 18 : Diamètres reconstruits en fonction des diamètres inventoriés

Ce dernier graphe nous permet d'observer l'intégralité des diamètres, et nous confirme que les algorithmes sous-évaluent les gros diamètres et qu'au contraire, les faibles diamètres sont sur-évalués.

5) Erreur sur les diamètres avec la distance.

On peut également observer la qualité de détection en fonction de la distance. Ainsi, on peut observer aussi bien l'erreur relative que l'erreur absolue.

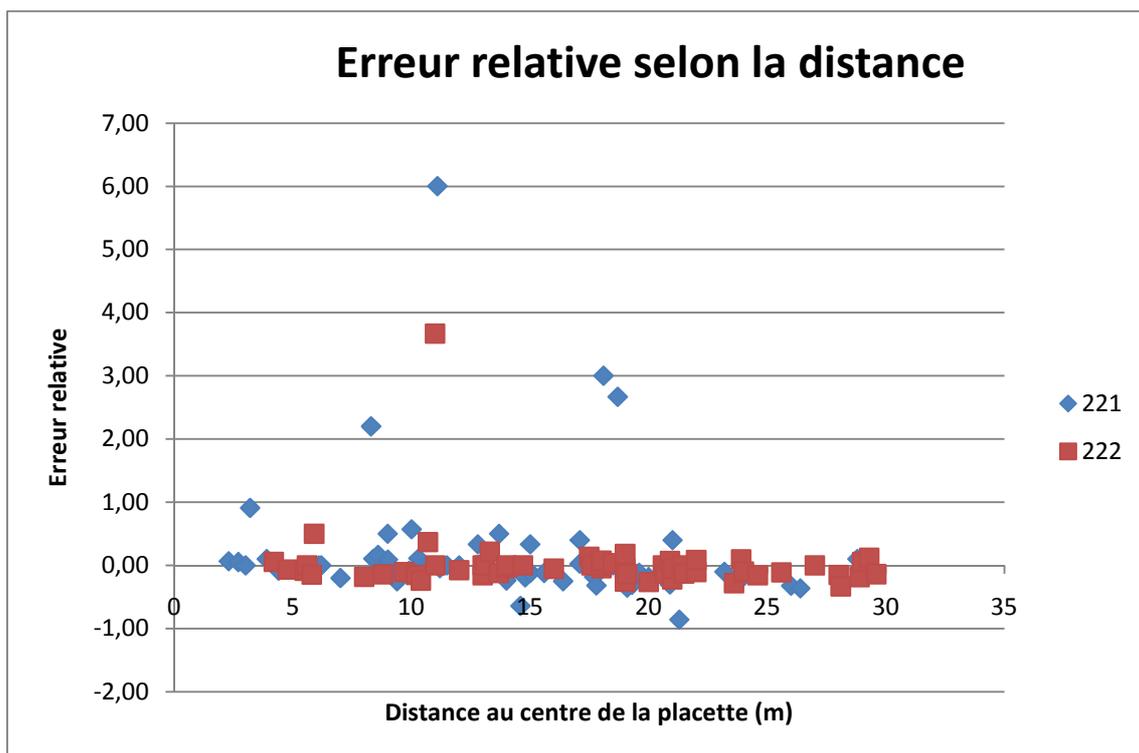


Figure 19: erreur relative en fonction de la distance

L'erreur relative est assez faible (l'erreur relative médiane sous-estime de 4%), ce qui n'empêche tout de même pas des cas où l'écart dépasse 100%. Cependant, toutes ces tiges ont un diamètre inventorié inférieur au diamètre de pré-comptage.

Tableau 8: erreur relative en fonction de l'intervalle de distance

Distance	1 à 5	5 à 10	10 à 15	15 à 20	20 à 25	25 à 30
Nombre de tiges	8	19	29	30	23	11
Erreur médiane	0,05	0,00	0,00	-0,02	-0,11	-0,14

Enfin, une analyse des erreurs relatives en fonction de la distance nous montre une tendance à la sous-évaluation avec la distance, qui n'est cependant pas clairement définie. En effet, on obtient une

erreur médiane plus élevée à grande distance, mais il y a beaucoup moins de tiges présentes, ce qui pourrait créer un effet levier de sous-évaluation.

Désormais, il s'agit d'estimer les erreurs de reconstruction.

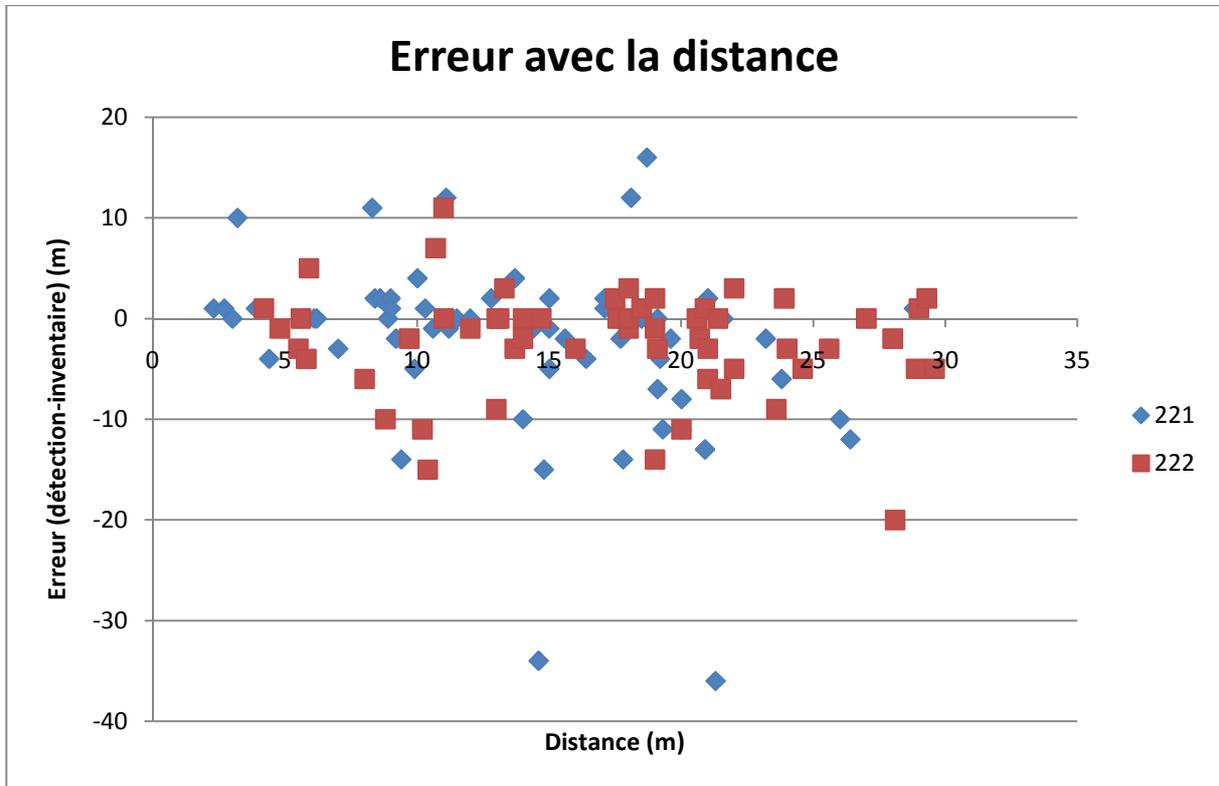


Figure 20: erreur avec la distance

L'erreur absolue moyenne sur le diamètre est de 2,38 cm de sous-évaluation. On peut observer également ce qu'il se passe en termes de classes de distance.

Tableau 9: erreur absolue en fonction de l'intervalle de distance

Distance	1 à 5	5 à 10	10 à 15	15 à 20	20 à 25	25 à 30
Nombre de tiges	8	19	29	30	23	11
Erreur médiane	1,00	0,00	0,00	-0,50	-3,00	-3,00

De la même manière que précédemment, on ne peut conclure sur l'impact de la distance sur la précision.

V. Conclusion et perspectives.

1) Résultats.

L'étude menée précédemment sur les deux placettes testées nous permet de tirer du travail effectué durant l'intégralité de ce stage certaines conclusions sur le logiciel COMPUTREE.

Dans un premier temps, la détection des tiges s'avère excellente : seules 5 tiges dont le diamètre est supérieur au diamètre de pré-comptage ne sont pas détectées, et principalement par manque de visibilité à travers de feuillage ou le sous-étage présent. Le taux de reconstruction des tiges de diamètre supérieur à 17,5 cm est de 93%.

L'aspect de localisation des tiges, qui reste très important dans le cadre de la reconstruction d'un inventaire forestier, est au point. En effet, nous sommes capables, à l'heure actuelle, de repositionner la quasi-intégralité des tiges inventoriées par simple tranchage des arbres reconstruits.

La reconstitution des diamètres est également de bonne qualité :

- Nous sommes capables de retrouver les classes de diamètre de manière assez précise.
- La précision est majoritairement bonne et ce malgré quelques cas très problématiques, qui ne concernent cependant que des tiges sous 17,5 cm de diamètre.
- Les tiges comprises entre 17,5 cm et 47,5 cm de diamètre sont quant à elles très bien reconstruites, le problème vient plutôt des gros diamètres.

Enfin, le résultat le plus intéressant concerne l'aspect temporel :

- Le temps de déplacement est quelque peu avec le TLS (compte-tenu de la masse à déplacer plus importante), l'écart est proche de la minute.
- Le temps d'acquisition des données est bien plus faible par le TLS (dans le cas de nos deux placettes, un scan durait 30 minutes, pour obtenir une résolution moyenne, qui n'a finalement pas été utilisée, mais on peut envisager de très bons résultats avec des scans de faible résolution, dont la durée est proche de 7 minutes) que lors d'un inventaire effectuée de manière plus classique (de 45 minutes à 2 heures selon que l'inventaire est réalisé à surface/angle fixe ou en plein)
- Le temps de traitement « bureau » est inférieur à cinq minutes.

L'un des gros avantages de cette méthode est de conserver les données, ce qui fait qu'une analyse des données peut être effectuée ailleurs que sur le terrain. De plus, la mesure effectuée ne fait intervenir aucun facteur humain, dans le sens où l'opérateur en charge du scanner n'a qu'à bien positionner le scanner laser et son trépied pour obtenir une mesure efficace. C'est pourquoi les biais actuellement issus de l'agent sont dépassés. La conservation des données permet également de pouvoir réévaluer les résultats obtenus en cas de modification d'algorithmes.

Enfin, on obtient une carte de la placette à un instant donné, ce qui est très utile pour effectuer un retour en cas d'oubli.

A l'heure actuelle, nous sommes donc capable de reconstruire un inventaire à surface fixe quasi-complet sur 25 mètres de rayon.

2) Si c'était à refaire...

Ce projet, de par la liberté relative d'action qui m'a été permise, m'a vraiment intéressé, et ce principalement par la découverte de nouvelles technologies appliquées dans un domaine où quasiment rien n'avait été effectué jusqu'à présent.

Je parle de liberté relative non pas au sujet des exigences du projet, mais plutôt dans le sens où j'aurais souhaité avoir les connaissances en programmation pour pouvoir encore mieux maîtriser le sujet et lui apporter encore plus.

Ainsi, si c'était à refaire, je pense qu'une intense période de formation en programmation serait nécessaire.

Ensuite, le fait de ne pas disposer d'un nombre suffisant de données propres est assez problématique, à savoir que les analyses effectuées ne portent que sur deux placettes d'un même massif forestier, ce qui fait que l'évaluation ne peut être réellement pertinente, et qu'une étude statistique à grande échelle est inenvisageable (et ce malgré les résultats évalués comme cohérents par Jean-Pierre Renaud). Ainsi, un temps bien supérieur de mise à disposition du scanner terrestre serait nécessaire.

3) Perspectives.

Dans un premier temps, il est nécessaire de mener ce projet à son terme, à savoir la reconstruction la plus précise possible des arbres ainsi que l'obtention des volumes.

Dans un premier temps, il s'agit de déterminer les raisons pour lesquelles un biais apparaît sur les gros diamètres. Comme on a pu le voir précédemment, le biais lié à la distance n'est pas évident. Il s'agit donc de travailler sur de nouvelles pistes :

- Est-ce dû au moindres carrés qui, liés au fait que les points aux bords des arbres sont très nombreux, fait que l'on sous-estime les diamètres ?
- Est-ce dû à un effet essence ?

La reconstruction, comme nous l'avons vu précédemment, est assez efficace, mais manque toutefois de précision dans certains cas, et en particulier l'étape de raccrochage est à améliorer. A l'heure actuelle, elle utilise des vecteurs, mais certains arbres éloignés parfois d'une très grande distance sont considérés comme identiques. L'utilisation des courbes de Bézier (rappel sur les courbes de Bézier en annexe) semble prometteuse, selon l'approche suivante :

- Supposons deux arbres numériques détectés, l'un étant juste au-dessus de l'autre, et qui ne sont pas considérés comme identiques. On détermine, pour ces deux tiges, les courbes de Bézier d'ordre 3 qui leur correspondent.
- On évalue, pour la courbe de la tige la plus basse, le t (supérieur à 1) nécessaire pour obtenir un point à une hauteur correspondant au Z minimal de l'arbre le plus haut.
- On évalue, pour la courbe de la tige la plus haute, le t (inférieur à 0) nécessaire pour obtenir un point à une hauteur correspondant au Z maximal de l'arbre le plus bas.
- On évalue alors, pour ces deux points, la distance à la courbe visée. Si les deux distances sont inférieures à une limite fixée, on considère que ces deux arbres sont identiques, si ce n'est pas le cas, les deux arbres sont différents.

L'avantage de cette méthode est qu'elle est localisée dans l'espace : le premier et le dernier point de la courbe de Bézier sont toujours l'un des points d'une des extrémités des arbres. Les erreurs de raccrochage sont ainsi limitées.

L'obtention des volumes est très aisément envisageable, sachant que nos arbres sont en réalité un empilement de cercles reconstruits. Il suffit dès lors d'évaluer le volume des petits cônes tronqués (le diamètre d'une couche à l'autre varie, la forme obtenue n'est pas parfaitement cylindrique) détectés pour obtenir les volumes souhaités.

La détection d'essences actuellement la donnée qui manque à la reconstruction complète, cette dernière, combinée aux caractéristiques du TLS, pourraient faire gagner un temps considérable à la mesure de données sur les placettes. Plusieurs pistes peuvent être évoquées :

- Fréquences des valeurs de réflectance caractéristique pour une essence ?
- Rugosité (en réflectance ou non) caractéristique d'une essence ?
- Utilisation de l'entropie du nuage de points pour distinguer les essences ?

La prise en compte du terrain pourrait améliorer les résultats de mesure. En effet, on a travaillé jusqu'à présent à une hauteur particulière du scanner, ce qui n'est pas complètement représentatif de la hauteur de poitrine. Ainsi, l'obtention du modèle numérique de terrain pourrait permettre d'aboutir à de meilleurs résultats, en termes de précision.

Enfin, durant la totalité de ce projet, nous avons utilisé exclusivement des méthodes générales et non spécifiques à la forêt. Il est possible, par exemple pour détecter les essences, d'observer l'orientation des bourgeons (alternes, opposés, hélicoïdaux), les variations de cylindricité (hêtre vs charme).

Bibliographie

[1] Bock et al, Les structures archéologiques et les peuplements de la forêt domaniale de Haye analysés par laser aéroporté, Rendez-vous Techniques, ONF, n°20, 2008, pp 15-18.

[2] Bienert et al, Tree detection and diameter estimation s by anaylsis of forest terrestrial scanner laser point clouds, IAPRS, volume 36, 2007.

[3] Mathworks [en ligne], disponible sur <http://www.mathworks.co.uk/matlabcentral/fileexchange/22684-ellipse-fit-direct-method>, consulté en 2010.

[4] Guil and Zapata, Lower order circle and ellipse Hough transform, Pattern Recognition Vol 30, n°10, pp. 1729-1744, 1997.

[5] Zelniker et al, A statistical analysis of the Delogne-Kasa method for fitting circles, Digital Signal Processing, n°16, pp 498-522, 2006.

Annexe 0 : termes techniques.

Bois fort-tige : arbre vivant dont le diamètre à 1,30 m de hauteur est supérieur ou égal à 7,5 cm.

Volume total : le volume total d'un arbre correspond au volume de la tige principale de l'arbre, auquel on ajoute le volume total de son houppier et de son système racinaire.

Capital sur pied : *surface terrière* d'un peuplement donné, qui est ramené à la parcelle ou à l'hectare.

Surface terrière : surface de la tranche à 1,30 m de hauteur d'un arbre, ramenée à l'hectare.

Ex : une tige de DBH = 30cm a une surface terrière $G = (\pi/4) * DBH^2 = 0,07m^2/ha$.

Itinéraire sylvicole : méthodes à mettre en œuvre pour obtenir le meilleur rendement possible.

Régénération : un peuplement en régénération présente un capital inférieur à $7m^2/ha$ et un diamètre moyen des tiges inférieur à 7,5 cm. La différence avec le taillis simple est que les tiges sont franc-pied (tronc propre) et n'appartiennent pas à des cépées (petites tiges regroupées sur un même pied).

Peuplement irrégulier : un peuplement irrégulier contient plusieurs essences avec une structuration horizontale (petits bois, bois moyens et gros bois) significative et une structuration verticale (arbres dominants, dominés, perche) marquée.

DBH : Diameter Breast Height, diamètre à 1,30m de hauteur.

L'estimation de la surface terrière s'effectue généralement non pas par une estimation du diamètre à 1,30 m mais par l'intermédiaire du relascope de Bitterlich (voir Annexe II).

MNT : Modèle Numérique de Terrain, ou représentation de la topographie d'une zone terrestre. Les données utilisées sont des coordonnées x,y,z des points relevés, ainsi que leur réflectance. Le MNT peut nous donner une vue de synthèse du terrain. Le travail sur ces données s'effectue par l'intermédiaire de Systèmes d'Information Géographiques (SIG), tels ArcMap ou GRASS.

Station totale : niveau laser à visée optique.

RMSE : Root Mean Square Estimation, pour résolution en termes des moindres carrés.

MLE : Maximum Likelihood Estimator, pour estimation par le maximum de vraisemblance.

Annexe I : Exemple de réseau de placettes permanentes.

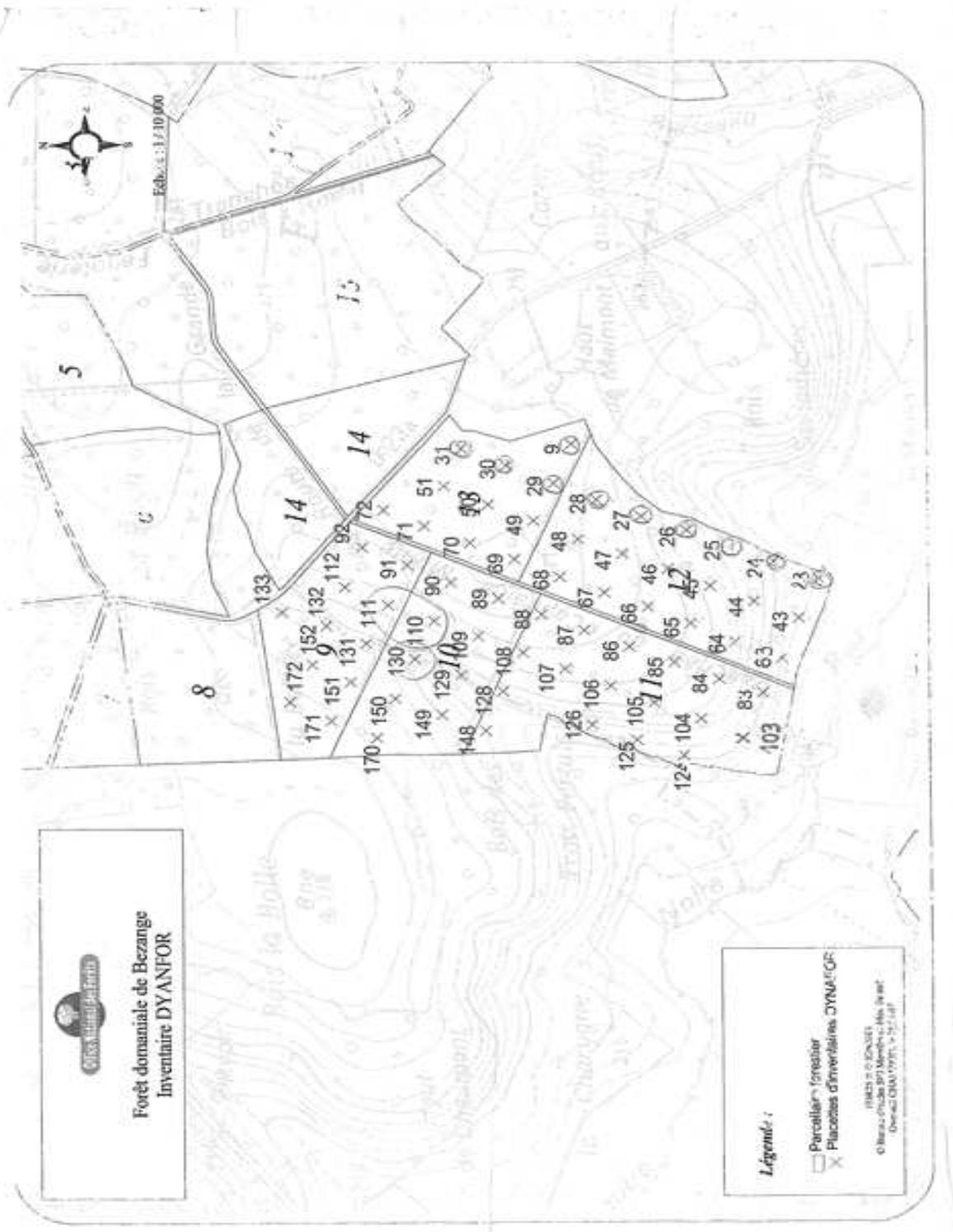


Figure 21: réseau de placettes permanentes de la forêt de Bezange (54)

Annexe II : méthode d'inventaire relascopique.

La méthode de la chaînette relascopique est utilisée afin d'estimer très rapidement le capital disponible sur un point particulier. Le relascope présente généralement 3 encoches : $\frac{1}{2}$, 1 et 2. On observe alors à travers ces encoches la tige considérée : si l'arbre est plus gros que l'encoche, alors il a au moins cette surface terrière, sinon, il n'est pas compté, sa surface terrière est considérée comme nulle. Enfin, si l'arbre est tangent, sa surface terrière est égale à la moitié.

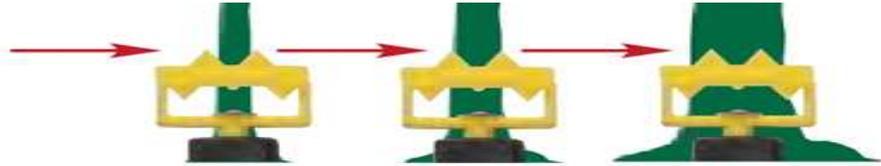


Figure 22: Principe du relascope pour une encoche de valeur 1

A gauche : $G = 0\text{m}^2/\text{ha}$.

Au milieu : $G = 0,5\text{m}^2/\text{ha}$.

A droite : $G = 1\text{m}^2/\text{ha}$.

Dès lors, on peut estimer, autour de ce point, le capital d'une parcelle ou d'une placette.

Lorsque l'on travaille sur des placettes à angle fixe, le relascope permet de considérer si l'on tient compte ou non de la tige visée dans l'inventaire. En effet, si son observation au travers du relascope fait que la tige présente un diamètre apparent inférieur à l'encoche, l'arbre ne sera pas inventorié. À l'inverse, si son diamètre apparent est supérieur à l'encoche, il sera inventorié.

Annexe III : Carte du réseau de placettes de la forêt de Bezange.

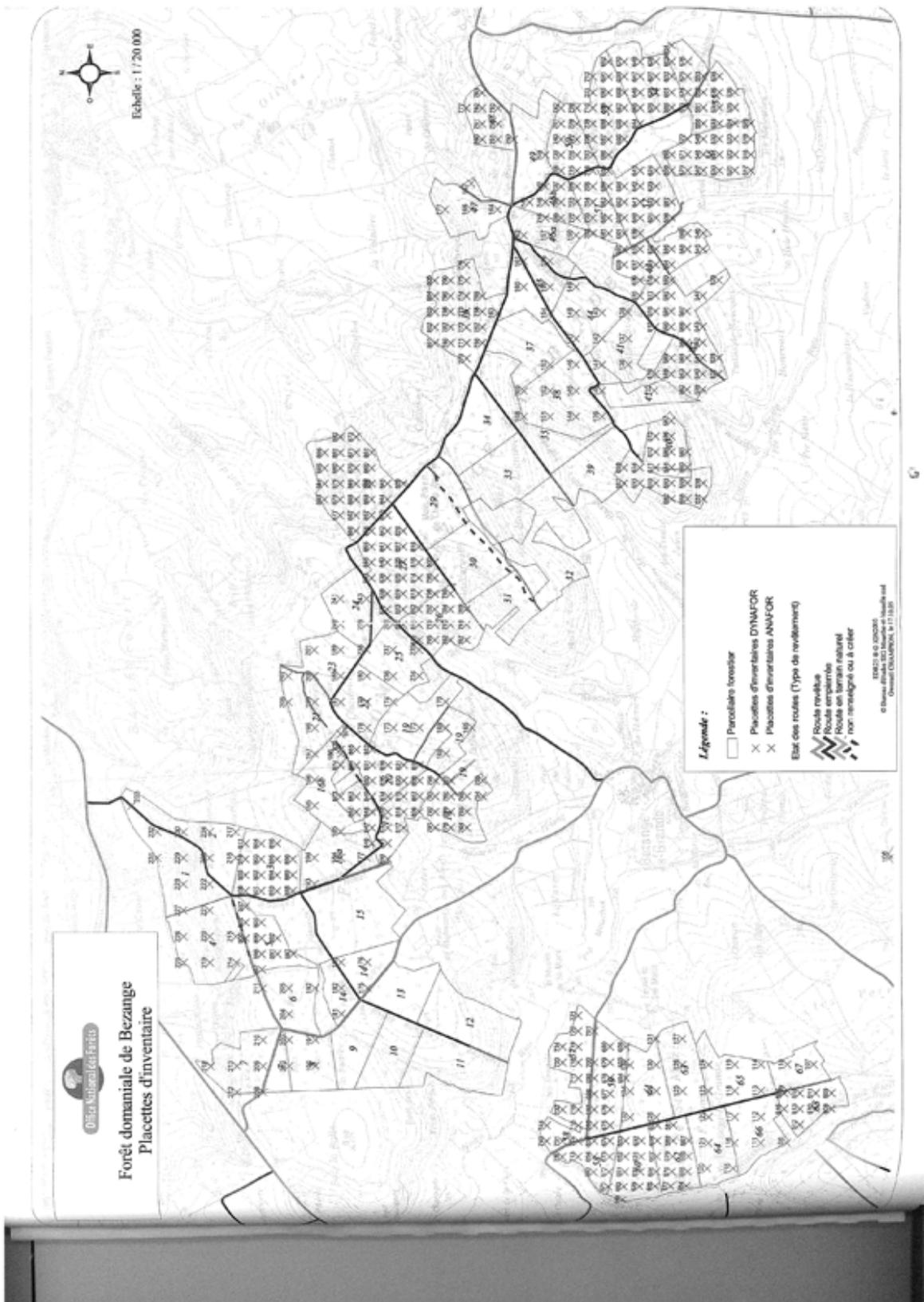


Figure 23: Carte de la forêt de Bezange (54)

Annexe IV : Protocole d'expérimentation en forêt de Bezange.

I. Objectif de l'expérience.

Confronter les mesures d'inventaire classique (diamètre/localisation/essence) avec les données obtenues avec le scanner laser terrestre Lidar, traitées par les algorithmes créés, afin d'évaluer leur validité.

II. Caractéristiques de l'expérience.

Cette expérience est effectuée sur deux placettes permanentes, possédant un peuplement à densité variable selon la direction et varié en termes de diamètres et d'essences.

Les données d'inventaires datant de plusieurs années, une nouvelle mesure s'avère nécessaire.

Afin de pouvoir rendre plus précise cette mesure, il a été convenu d'évaluer les diamètres à une hauteur fixe, obtenue par l'intermédiaire d'un niveau laser. Cette hauteur sera alors représentée sur les arbres par un trait de peinture, ce qui permettra une mesure manuelle plus précise qu'à l'accoutumée, ainsi qu'une comparaison plus aisée avec les données obtenues par le scanner.

Enfin, afin d'évaluer l'efficacité en distance du scanner laser terrestre et en diamètres des algorithmes, il a été convenu de procéder cette opération sur l'ensemble des tiges situées sur un rayon de 30 m autour du centre de ces placettes.

III. Localisation de l'expérience.

Forêt domaniale de Bezange, Meurthe et Moselle (54), parcelle 4, placette 221 et parcelle 1, placette 222.

IV. Matériel nécessaire.

- Laser (portée minimale de 20m)
- Scanner laser terrestre (avec accessoires)
- Peinture (avec matériel)
- TruePulse (azimut, distance)
- Mètre ruban
- Compas forestier
- Gabarit

V. Protocole.

- 1) Mise en place du niveau laser.
- 2) Tracé du pourtour du DBH.
- 3) Tracé de l'arc de cercle.
- 4) Localisation de l'arbre.
- 5) Mesure du DBH au compas forestier.
- 6) Mesure de la circonférence.
- 7) Mesure de la portion d'arc visible
- 8) Travail des données terrain obtenues
- 9) Mesure de la placette au TLS.

VI. Description des étapes.

- 1) Mise en place du niveau laser.

Noter la hauteur de la tête d'émission du laser par rapport au sol, afin de mettre au même niveau le scanner terrestre lors de la suite de l'expérience.

- 2) Tracé du pourtour du DBH.

Repérer que l'arbre est touché par le rayon du laser.

Numéroter cet arbre.

Positionner le bas de la rubalise sur le tracé du rayon laser.

Peindre le tour de la rubalise à la bombe de peinture.

Laisser sécher la peinture.

- 3) Tracé de l'arc de cercle.

Retirer la rubalise.

Tracer les bords droits et gauche de l'impact du laser sur le tronc avec une cale à la bombe de peinture.

Laisser sécher la peinture.

- 4) Localisation de l'arbre.

Localiser l'arbre via le true Pulse, à la moitié de la hauteur de la rubalise.

- 5) Mesure des DBH au compas forestier.

Mesurer deux diamètres de l'arbre; à la moitié de la hauteur de la rubalise, le premier dans l'axe du laser, le second orthogonalement.

6) Mesure de la circonférence

Mesurer au mètre ruban la circonférence de l'arbre, à la moitié de la hauteur de la rubalise.

7) Mesure de la portion d'arc visible.

Mesurer au mètre ruban la portion de l'arc visible, repérée précédemment à la peinture.

8) Travail des données terrain obtenues.

9) Mesure de la placette au TLS.

VII. Planning de l'expérience.

Les étapes 1 à 7 seront effectuées les 15 et 16 avril 2010, respectivement sur les placettes 221 et 222.

L'étape 8 sera effectuée après l'acquisition complète des données.

L'étape 9 sera effectuée à une date non encore définie, mais ne va durer qu'une journée au maximum.

Remarque importante : si le temps mis pour effectuer les placettes 221 et 222 est inférieur à une journée, nous effectuerons deux mesures supplémentaires sur deux placettes, la 239 et la 240, parcelle 24.

Annexe V : interface du logiciel COMPUTREE développé en JAVA.

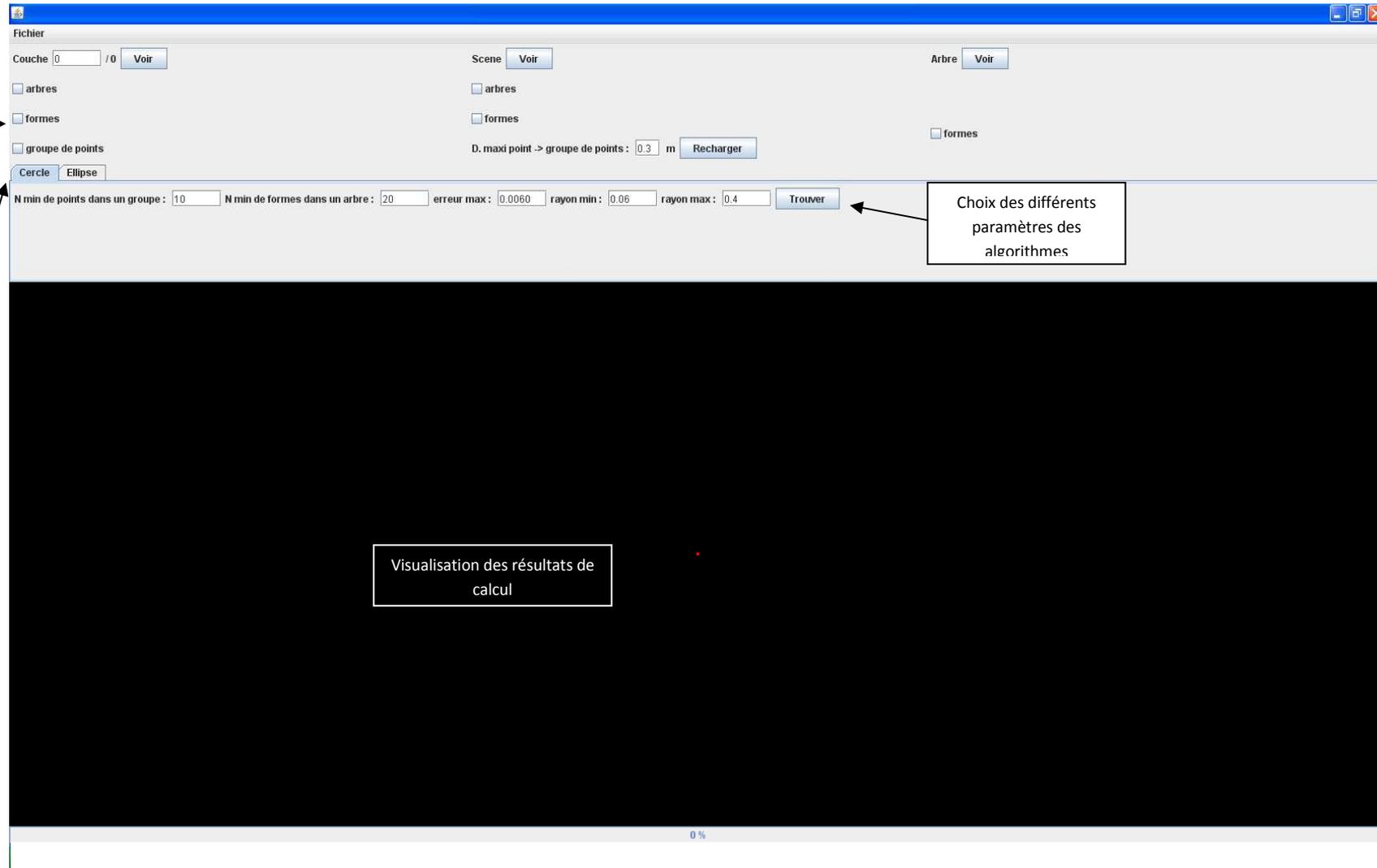


Figure 24: Interface utilisateur de COMPUTREE développé sous JAVA

Annexe VI : Algorithme de calcul des diamètres des arbres reconstruits sur la version JAVA.

```

## répertoire de travail
RepData = "data/StageEtienne/ScanVSmesure/Tests/"
#####
##### Paramètres liés à la perche mesurée#####
#####
#####

## Nom des fichiers issus de l'algorithme et de l'inventaire
NomFichierAlgo = "6p7g.txt"

## Lecture du jeu de cercles issus de l'algorithme
data=read.table(paste(RepData, NomFichierAlgo,sep=""),sep=";",dec=".",na="",header=T)
names(data) = c("Arbre", "X", "Y", "Z", "R", "Error")

## Mise en tableau des données utiles.
M = matrix(c(data$Arbre,data$X,data$Y,data$Z,data$R),ncol = 5)

## Tableau d'identification des arbres
N = matrix(1:(max(M[,1]+1)))

## Calcul des coordonnées et du nombre de cercles des arbres reconstruits
## Création des tableaux représentant la somme des X et Y pour la placette puis ramenée à chaque arbre
A = matrix(nrow=max(N),ncol = 2)
B = matrix(nrow=max(N),ncol = 2)

## Calcul de la somme des X et Y pour la placette
c <- 0
d <- 0
j <- 1
for(i in 1:(max(N)-1)){
  while(M[j,1]<i){
    c <- c + M[j,2]
    d <- d + M[j,3]
    j <- j+1}
  A[i,1] = c
  A[i,2] = d}
A[max(N),1] = sum(M[,2])
A[max(N),2] = sum(M[,3])

## Calcul de la somme des X et Y ramenée à chaque arbre.
for (i in 2:(max(N))){
  B[i,1] = A[i,1]-A[(i-1),1]
  B[i,2] = A[i,2]-A[(i-1),2]
}

B[1,1] = A[1,1]
B[1,2] = A[1,2]

## Calcul du nombre de cercles par arbre
## Création des tableaux donnant la somme des cercles par placette, puis ramenée à chaque arbre
C = matrix(nrow = max(N))
D = matrix(nrow = max(N))

## Calcul de la somme des cercles pour la placette
c <- 0
j <- 1
for (i in 1:(max(N)-1)){
  while(M[j,1]<i){
    c <- c+1
    j <- j+1}
  C[i,1] = c}
C[max(N),1] = length(M[,1])

## Résultat final
for (i in 2:max(N)){
  D[i,1] = C[i,1]-C[(i-1),1]
}
D[1,1] = C[1,1]

## Création du tableau donnant la moyenne en X et Y pour chaque arbre.
E = matrix(nrow = max(N),ncol = 2)

```

```

## Calcul de X et Y moyen pour chaque arbre.
for (i in 1:max(N)){
E[i,1] = B[i,1]/D[i,1]
E[i,2] = B[i,2]/D[i,1]
}

## Calcul d'une tranche de 50cm autour de 1.3m par rapport au minimum de l'arbre
## Définition du tableau donnant le Z minimal par arbre
F = matrix(nrow = max(N))

## Calcul du Z minimal par arbre
for (i in 1:(max(N)-1)){
F[i+1,1] = M[C[i,1]+1,4]}
F[1,1] = M[1,4]

## Création du tableau donnant une tranche de 50cm autour du 1.30m de l'arbre
G = matrix(nrow = max(N), ncol = 2)

## Calcul du Z minimal et du Z maximal de la tranche
G[,1] = F[,1]+1.3-0.25
G[,2] = F[,1]+1.3+0.25

## Calcul du diamètre à 1.30m
H = matrix(ncol=2,nrow=0)
for (i in 1:nrow(N)) {
M2 = M[M[,1]==i-1,]
Dmin = G[i,1]
Dmax = G[i,2]
somme = 0
compteur = 0
for (j in 1:nrow(M2)) {
if (M2[j,4]>=Dmin && M2[j,4]<=Dmax) {
somme = somme + M2[j,5]
compteur = compteur + 1
}
}
if (compteur>0) {
H = rbind(H, c(i,somme/compteur))
} else {
H = rbind(H, c(i,0))
}
}

## Création d'un tableau comportant le numéro d'arbre, ses coordonnées X et Y, son diamètre à 1.30m
K = matrix(c(H[,1],E,2*H[,2]),nrow=max(N))

## Travail sur les coordonnées angulaires et en distance
## Définition de la fonction distance
Distance <- fonction(x,y){
distance = sqrt(x*x+y*y)
return(distance)}

## Définition du tableau donnant la distance de chaque arbre avec l'origine
L = matrix(ncol=1,nrow=max(N))

## Calcul de la distance entre chaque arbre et l'origine.
for (i in 1:max(N)){
L[i,] = Distance(E[i,1],E[i,2])
}

## Définition du tableau des azimuts
O = matrix(ncol=1,nrow=max(N))

## Calcul de l'azimut de chaque arbre
for (i in 1:max(N)){
O[i,1] = ifelse((E[i,1]/L[i,])>0,ifelse((E[i,2]/L[i,])>0,asin(E[i,1]/L[i,])*180/pi,acos(E[i,2]/L[i,])*180/pi),ifelse((E[i,2]/L[i,])>0,asin(E[i,1]/L[i,])*180/pi,-
acos(E[i,2]/L[i,])*180/pi))
O[i,1] = O[i,1]*400/360
O[i,1] = ifelse(O[i,1]<0,O[i,1]+400,O[i,1])
}

## Création d'un tableau d'infos complètes sur les arbres
P = matrix(ncol=6,nrow=max(N))

## Tableau comportant le numéro d'arbre, ses coordonnées X et Y, sa distance, son azimut, son diamètre à 1.30m
for (i in 1:max(N)){
P[i,] = c(N[i,],K[i,2],K[i,3],L[i,],O[i,],K[i,4])}

```

Annexe VII : évaluation des paramètres de COMPUTREE.

Tableau 10: Tableau d'évaluation des paramètres de COMPUTREE

Essai	Séparation en couches		Cercles			Filtre	Résultats		
	Distance	Epaisseur	Erreur max	Rayon Min	Rayon Max	Nb cercles	DBH	Cercles	Arbres
1	30	1	0,006	6	40	20	0,127589	61	1
2	50	1	0,006	6	40	20	0,127589	61	1
3	100	1	0,006	6	40	20	0,127589	55	1
4	200	1	0,006	6	40	20	0,134062	28	1
5	10	1	0,006	6	40	20	0,146207	44	1
6	20	1	0,006	6	40	20	0,128573	62	1
7	15	1	0,006	6	40	20	0,127913	63	1
8	12	1	0,006	6	40	20	0,135312	60	1
9	13	1	0,006	6	40	20	0,151282	66	1
10	15	5	0,006	6	40	20	0,000000	0	1
11	30	5	0,006	6	40	20	0,000000	0	1
12	50	5	0,006	6	40	20	0,000000	0	1
13	15	2	0,006	6	40	20	0,129303	25	1
14	30	2	0,006	6	40	20	0,129303	22	1
15	50	2	0,006	6	40	20	0,129303	22	1
16	15	1	0,001	6	40	20	0,000000	0	1
17	30	1	0,001	6	40	20	0,000000	0	1
18	50	1	0,001	6	40	20	0,000000	0	1
19	15	1	0,01	6	40	20	0,124040	137	2
20	30	1	0,01	6	40	20	0,123643	131	2
21	50	1	0,01	6	40	20	0,123643	131	2
22	15	1	0,05	6	40	20	0,123747	141	2
23	30	1	0,05	6	40	20	0,128624	140	3
24	50	1	0,05	6	40	20	0,128689	136	3
25	15	1	0,1	6	40	20	0,123747	141	2
26	30	1	0,1	6	40	20	0,128803	140	3
27	50	1	0,1	6	40	20	0,128689	136	4
28	15	1	0,01	1	40	20	0,173050	1056	4
29	30	1	0,01	1	40	20	0,113322	826	5
30	50	1	0,01	1	40	20	0,113322	1062	4
31	15	1	0,01	10	40	20	0,000000	0	1
32	30	1	0,01	10	40	20	0,367150	22	2
33	50	1	0,01	10	40	20	0,270192	49	2
34	30	1	0,01	1	10	20	0,113322	826	5
35	50	1	0,01	1	10	20	0,113322	1062	2
36	30	1	0,01	1	100	20	0,113322	826	5
37	50	1	0,01	1	100	20	0,113322	1062	4
38	30	1	0,01	1	10	10	0,113322	826	4
39	50	1	0,01	1	10	10	0,113322	1062	3
40	30	1	0,01	1	5	20	0,0991070	431	3
41	50	1	0,01	1	5	20	0,0991070	662	1

Annexe VIII : algorithmes de comparaison des données développés sous R.

```

## Comparaison des deux fichiers
## T1 = comparaison en X et Y

T1 = matrix(nrow=N, ncol=k)
for (i in 1:N){
  for (j in 1:k){
    T1[i,j] = ifelse(M[i,2]<(X[j,1]+Xcor-0.5),0,ifelse(M[i,2]>(X[j,1]+Xcor+0.5),0,ifelse(M[i,3]<(Y[j,1]+Ycor-0.5),0,ifelse(M[i,3]>(Y[j,1]+Ycor+0.5),0,1))))
  }
}

## Nombre d'arbres détectés par arbre inventorié (en xy)
U1 = matrix(nrow=k,ncol=2)
for (i in 1:k){
  U1[i,1] = i
  U1[i,2] = sum(T1[,i])
}

## T2 = comparaison en azimut-distance
## Changement des coordonnées des points avec la correction
X = X + Xcor
Y = Y + Ycor
D = Distance(X,Y)

Az = matrix(nrow=k,ncol=1)
## Calcul de l'azimut de chaque arbre inventorié corrigé
for (i in 1:k){
  Az[i,1] = ifelse((X[i,]/D[i,])>0,ifelse((Y[i,]/D[i,])>0,asin(X[i,]/D[i,])*180/pi,acos(Y[i,]/D[i,])*180/pi),ifelse((Y[i,]/D[i,])>0,asin(X[i,]/D[i,])*180/pi,-acos(Y[i,]/D[i,])*180/pi))
  Az[i,1] = Az[i,1]*400/360
  Az[i,1] = ifelse(Az[i,1]<0,Az[i,1]+400,Az[i,1])
}

## Calcul de l'azimut où l'on ajoute le décalage
Az2 = matrix(nrow=k,ncol=2)

for (i in 1:k){
  Az2[i,1] = Az[i,1] + 10
  Az2[i,2] = Az[i,1] - 10
  Az2[i,1] = ifelse(Az2[i,1]>400,Az2[i,1]-400,Az2[i,1])
  Az2[i,2] = ifelse(Az2[i,2]<0, Az2[i,2]+400,Az2[i,2])
  Az2[i,2] = ifelse(Az2[i,1]<10,Az2[i,2]-400,Az2[i,2])
}

T2 = matrix(nrow=N,ncol=k)
for (i in 1:N){
  for (j in 1:k){
    T2[i,j] = ifelse(A[i,1]<(D[j,1]-1),0,ifelse(A[i,1]>(D[j,1]+1),0,ifelse(B[i,1]<(Az2[j,2]),0,ifelse(B[i,1]>(Az2[j,1]),0,1))))
  }
}

## Nombre d'arbres détecté par arbre inventorié
U2 = matrix(nrow=k,ncol=2)
for (i in 1:k){
  U2[i,1] = i
  U2[i,2] = sum(T2[,i])
}

K = matrix(c(U1,U2[,2]),nrow=k)
#####
## Définition de la distance entre 2 points de la scène
## Distance 2
Distance2 <- function(a,b){
  distance2 = sqrt(((M[a,2]-X[b,1])^2)+((M[a,3]-Y[b,1])^2))
  return(distance2)}

## Détermination de l'arbre détecté le plus proche pour tout arbre inventorié.
V = matrix(nrow=max(N),ncol=k)

```

```

for(i in 1:max(N)){
  for(j in 1:k){
    V[i,j] = Distance2(i,j)}

  ## Evaluation de la distance de l'arbre détecté le plus proche pour chaque arbre inventorié
  W = matrix(nrow=k,ncol=4)
  for(i in 1:k){
    W[i,1] = min(V[,i])
  }

  ## Recherche de l'arbre qui est le plus proche.
  Proche <- fonction(j){
    i <- 1
    while(V[i,j]>W[j,1]+0.1){
      i <- i+1}
    return(i)}

  ## Numéro de l'arbre détecté le plus proche pour tout arbre inventorié
  for (j in 1:k){
    W[j,2] = Proche(j)}

  ## Evaluation de la distance de l'arbre inventorié le plus proche pour chaque arbre détecté
  W1 = matrix(nrow=max(N),ncol=2)
  for(i in 1:max(N)){
    W1[i,1] = min(V[i,])
  }

  ## Recherche de l'arbre qui est le plus proche
  Proche2 <- fonction(i){
    j <- 1
    while(V[i,j]>W1[i,1]+0.1){
      j <- j+1}
    return(j)}

  ## Numéro de l'arbre inventorié le plus proche pour tout arbre détecté
  for (i in 1:max(N)){
    W1[i,2] = Proche2(i)}

  ## Test de concordance "arbre détecté le plus proche pour arbre inventorié" VS "arbre inventorié le plus proche pour chaque arbre détecté"
  for(j in 1:k){
    W[j,3] = W1[W[j,2],2]
    W[j,4] = ifelse((W[j,3]-j)>0,1,ifelse((W[j,3]-j)<0,1,0))
  }

```

Annexe IX : version finale de COMPUTREE.



Figure 25: Lecture du fichier de point

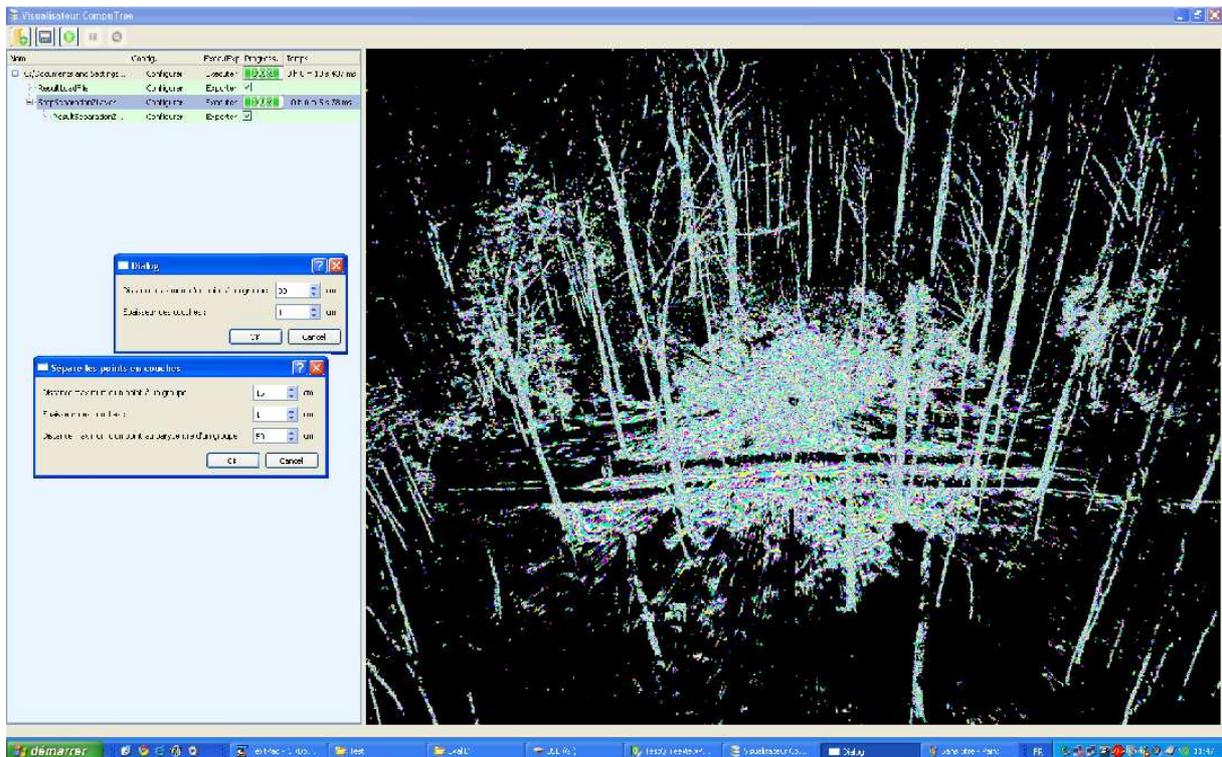


Figure 26: Séparation en couches et détection de formes

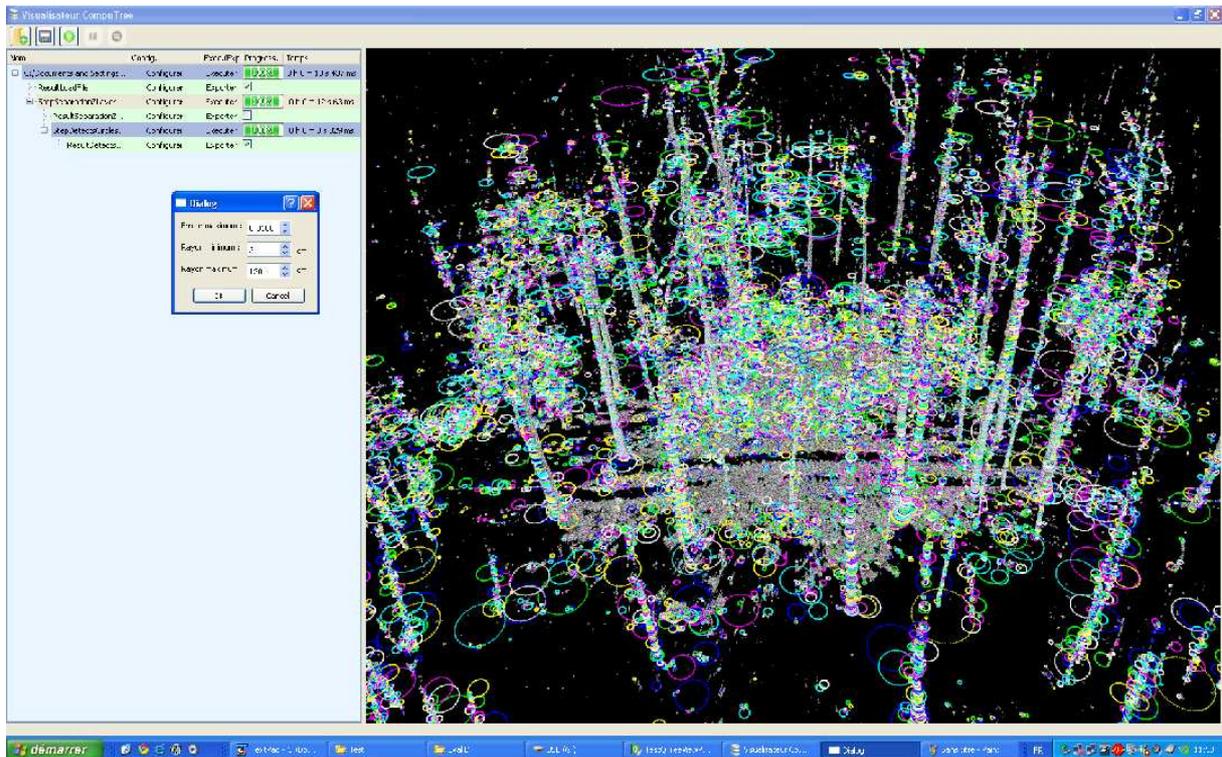


Figure 27: Détection des cercles

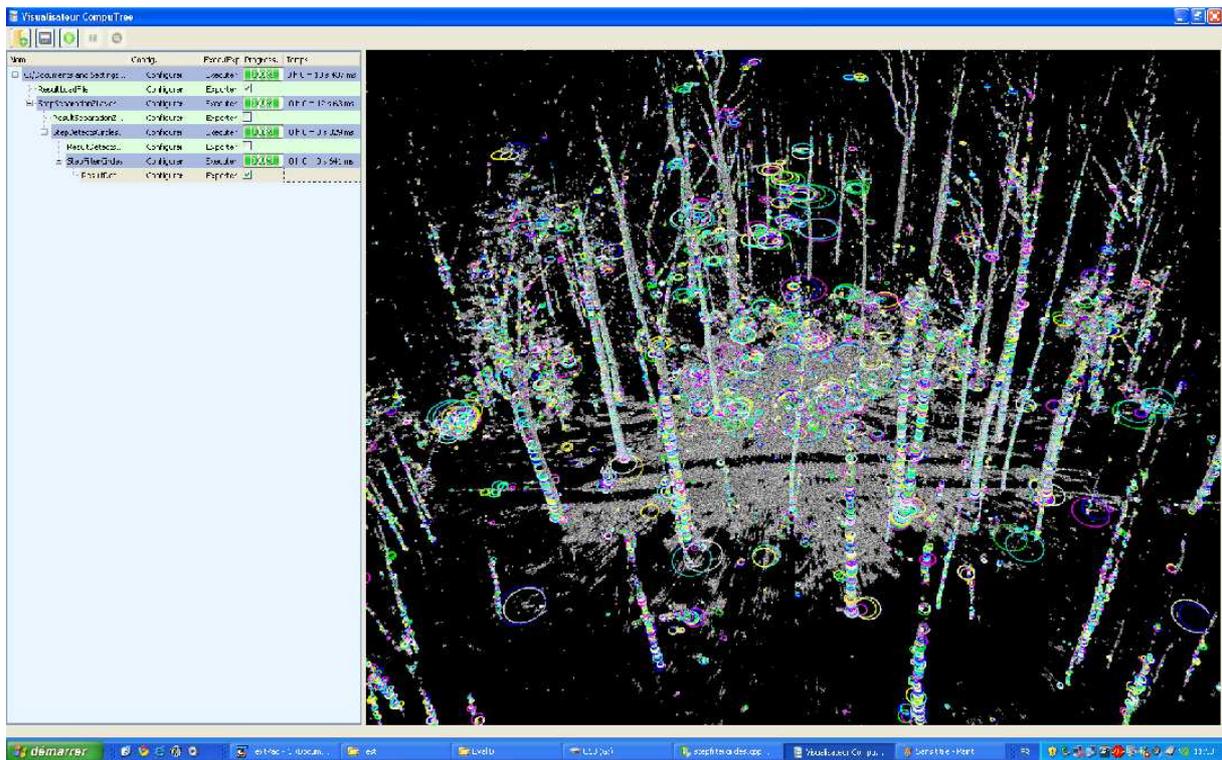


Figure 28: Pré-nettoyage des cercles

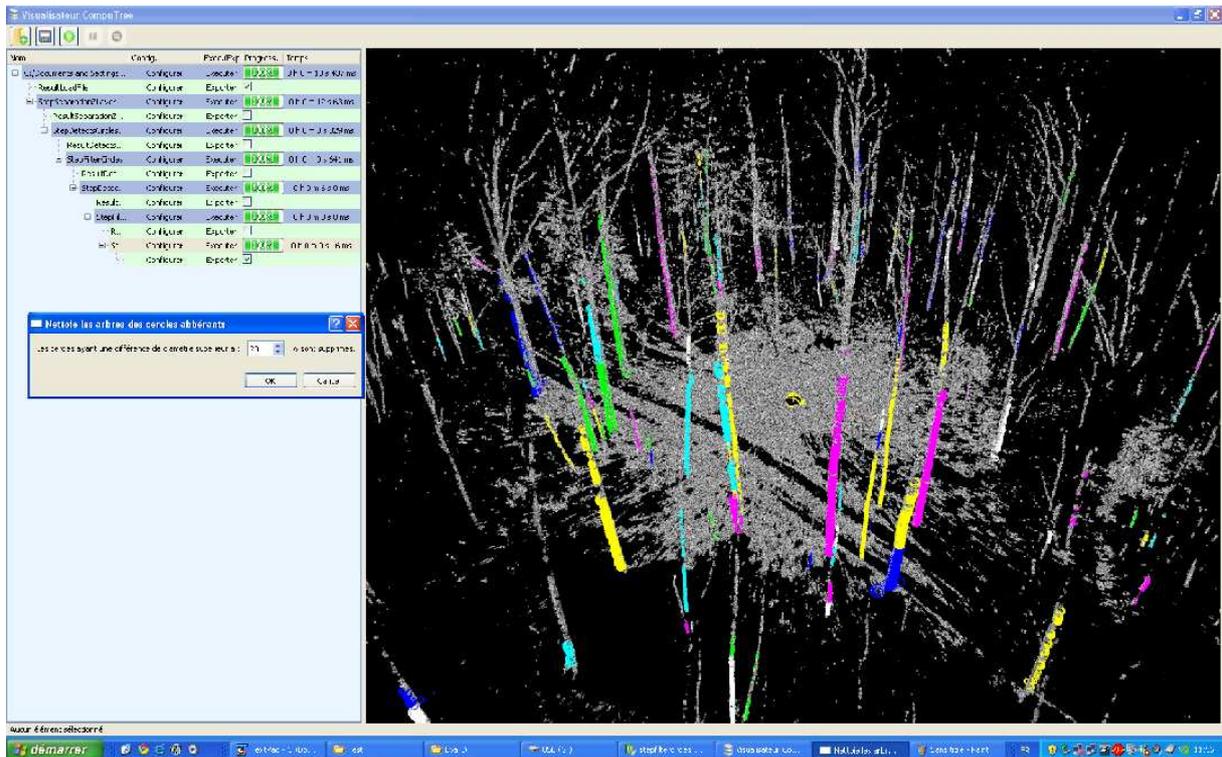


Figure 31: Nettoyage des arbres

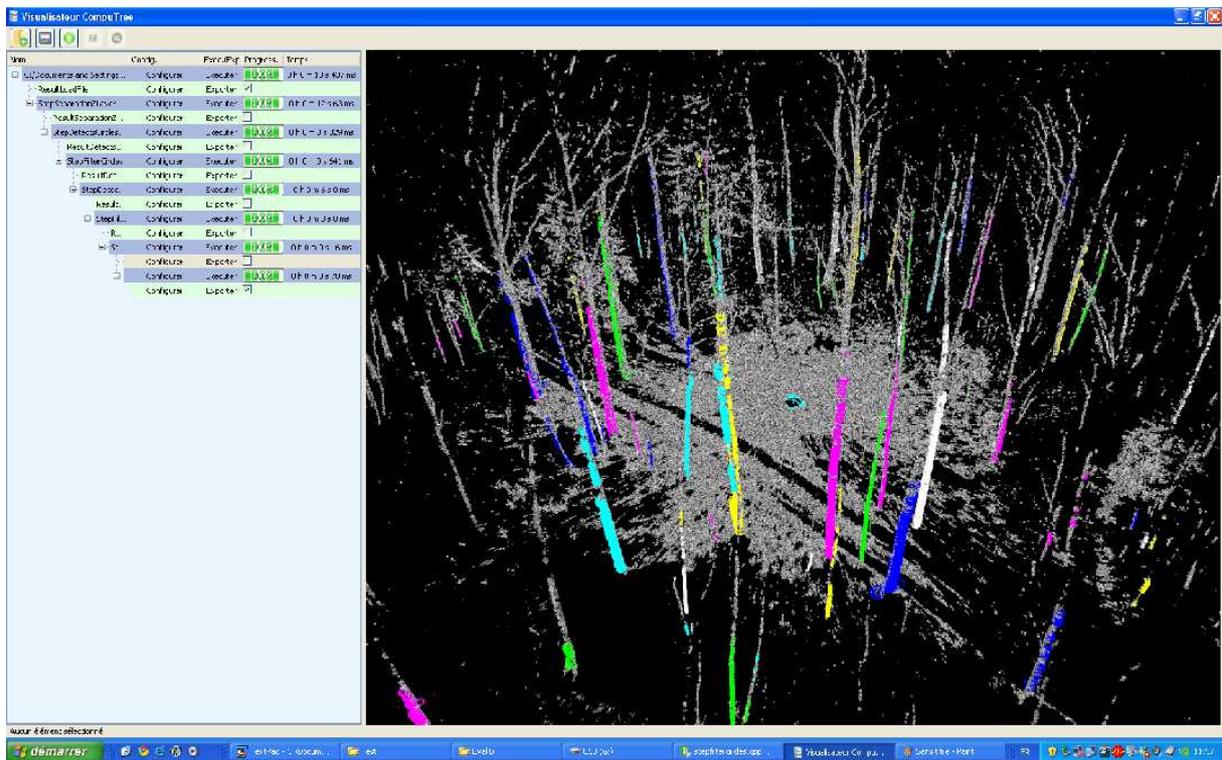


Figure 32: Raccrochage des arbres

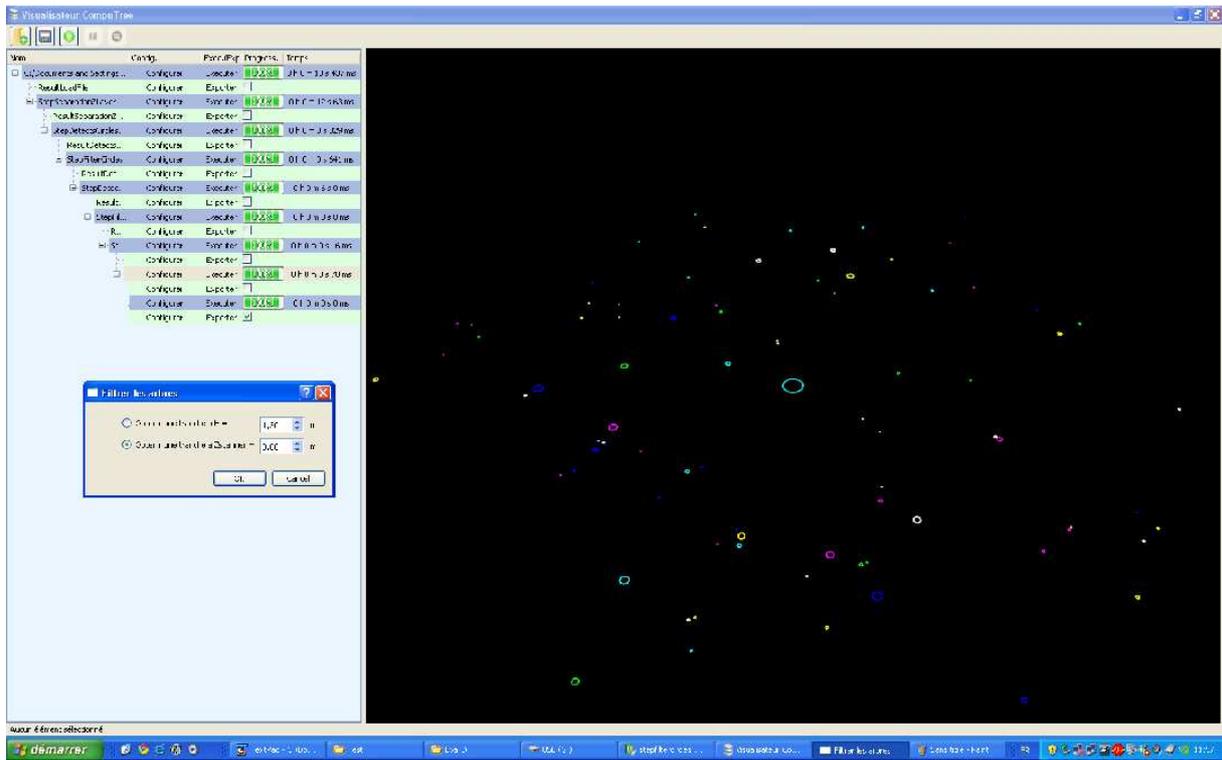


Figure 33: Reconstruction de tranches

Annexe X : Algorithme de localisation des arbres de la forêt de Bezange, développés sous R.

```

## répertoire de travail
RepData = "data/StageEtienne/ScanVSmesure/Bezange/221/"
#####
#####
#####
#####

## nom de l'inventaire terrain et des sorties de l'algorithme correspondant
NomFichierAlgo = "Tranche 221.txt"
NomFichierInv = "Inv 221.csv"
NomFichierArbre = "Arbre 221.txt"

## Définition des variables de correction en coordonnées et en angle
AngleCorrection = 40
Xcor = 0
Ycor = 0

## Lecture des cercles trouvés par l'algo
data=read.table(paste(RepData, NomFichierAlgo,sep=""),sep=";",dec=".",na="",header=T)names(data) = c("Arbre","X","Y","Z","R")

## Mise en tableau des données de l'algo(N°arbre,coordonnées,rayon)
M = matrix(c(data$Arbre,data$X,data$Y,data$Z,data$R),ncol = 5)

## Longueur du tableau
N = (length(M)/5)

## Définition de la fonction distance
Distance <- fonction(x,y){
distance = sqrt(x*x+y*y)
return(distance)}
#####
## Lecture du fichier d'inventaire
inv=read.table(paste(RepData, NomFichierInv,sep=""),sep=";",dec=".",na="",header=T)
names(inv) = c("Placette", "AzDeg", "Distance", "Num", "Essence", "C", "D1", "D2", "Dist2", "Inc", "Rem")

## Longueur du tableau
L = length(inv$Placette)

## Création d'un tableau comportant la circonférence de l'arbre, sa distance, son azimuth
M2 = matrix(c(inv$C,inv$Distance,(inv$AzDeg+inv$Inc)),ncol = 3)

## Nombre d'arbres inventoriés
k=length(M2)/3
o=length(M2)/3

## Tableau comportant le numéro de l'arbre inventorié, sa circonférence, sa distance, son azimuth
M3 = matrix(ncol=4,nrow=k)
for (i in 1:k){
M3[i,1]=i
}
M3[,2] = M2[,1]
M3[,3] = M2[,2]
M3[,4] = M2[,3]

## Calcul de l'angle corrigé des arbres inventoriés
M3[,4] = M2[,3]+AngleCorrection
M3[,4] = ifelse(M3[,4]<0,M3[,4]+360,ifelse(M3[,4]>360,M3[,4]-360,M3[,4]))

## Fonctions de calcul des coordonnées
XScanCoord <- fonction(az, d, maxAngle = 360){
  azrad <- ((az*2*pi)/maxAngle)
  x <- d*sin(azrad)
  return(x)
}

```

```

YScanCoord <- function(az, d, maxAngle = 360){
  azrad <- ((az*2*pi)/maxAngle)
  y <- d*cos(azrad)
  return(y)
}

## Initialisation des matrices de coordonnées xy des cercles de l'inventaire
X = matrix(nrow=k,ncol=1)
Y = matrix(nrow=k,ncol=1)

## Calcul des coordonnées XY des cercles inventoriés
for (i in 1:k){
  X[i,1] = XScanCoord(M3[i,4],M3[i,3])
  Y[i,1] = YScanCoord(M3[i,4],M3[i,3])
}

## création du graphe
dev.new(10,10)
par(pty="s")
plot(0,0,cex=0.01,xlim=c(-30,30),ylim=c(-30,30))

## tracé des cercles de repérage
symbols(0,0,circles = 1, inches = FALSE,fg="darkgrey",add=TRUE)
symbols(0,0,circles = 5, inches = FALSE,fg="darkgrey",add=TRUE)
symbols(0,0,circles = 10, inches = FALSE,fg="darkgrey",add=TRUE)
symbols(0,0,circles = 15, inches = FALSE,fg="darkgrey",add=TRUE)
symbols(0,0,circles = 20, inches = FALSE,fg="darkgrey",add=TRUE)
symbols(0,0,circles = 25, inches = FALSE,fg="darkgrey",add=TRUE)

## tracé des cercles inventoriés
for (i in 1:k){
  symbols(X[i,1]+Xcor,Y[i,1]+Ycor,circles =(2*M3[i,2]/(1000*pi)), inches = FALSE,fg="green",bg="green",add=TRUE)
}

K = matrix(nrow=k,ncol=2)
K[,1] = X
K[,2] = Y

#####
## Lecture du jeu de données contenant les arbres détectés
tree=read.table(paste(RepData, NomFichierArbre,sep=""),sep=";",dec=".",na="",header=T)
names(tree) = c("Arbre", "X", "Y", "Z", "R", "E")
a = (max(tree$Arbre)+1)

## Création des tableaux donnant la somme des cercles par placette, puis ramenée à chaque arbre
C = matrix(nrow = a)
D = matrix(nrow = a)

## Calcul de la somme des cercles pour la placette
c <- 0
j <- 1
for (i in 1:(a-1)){
  while(tree$Arbre[j]<i){
    c <- c+1
    j <- j+1}
  C[i,1] = c}
C[a,1] = length(tree$Arbre)

## Calcul du nombre de cercles par arbre
for (i in 2:a){
  D[i,1] = C[i,1]-C[(i-1),1]
}
D[1,1] = C[1,1]

## Calcul du point le plus bas de chaque arbre
D = matrix(nrow=a,ncol=2)
for(j in 1:(a-1)){
  B = matrix(nrow=0,ncol=5)
  for (i in C[j,1]:C[(j+1),1]){
    B = rbind(B,c(tree$Arbre[i],tree$X[i],tree$Y[i],tree$Z[i],tree$R[i]))}

```

```

D[(j+1),1] = j
D[(j+1),2] = min(B[,4])

B = matrix(nrow=0,ncol=5)
for (i in 1:C[1,1]){
B = rbind(B,c(tree$Arbre[i],tree$X[i],tree$Y[i],tree$Z[i],tree$R[i]))
D[1,1] = 0
D[1,2] = min(B[,4])

## Test du nombre d'arbres reconstruits à plus de 2.5m de haut
K = matrix(nrow=(length(D)/2),ncol=2)
for (i in 1:(length(D)/2)){
K[i,2] = ifelse(D[i,2]>2.5,1,0)
K[i,1] = i
}

## On supprime les arbres qui sont reconstruits au minimum à plus de 2,5 m
G = matrix(nrow=(length(D)/2),ncol=3)
G[,1] = D[,1]
G[,2] = D[,2]
G[,3] = K[,2]

G = G[order(G[,3],decreasing=TRUE),]
H = matrix(nrow=0,ncol=1)

i = 1
while(G[i,3]>0.5){
H = rbind(H,G[i,1])
i <- (i+1)
}

Z = M[-c(t((H+1))),]
N = (length(Z)/5)

## Tracé des points obtenus avec le scanner
#####
## Définition de la distance entre 2 points de la scène
## Distance 2
Distance2 <- function(a,b){
distance2 = sqrt(((Z[a,2]-X[b,1])^2)+((Z[a,3]-Y[b,1])^2))
return(distance2)}

## Détermination de l'arbre détecté le plus proche pour tout arbre inventorié.
V = matrix(nrow=N,ncol=k)
for(i in 1:N){
for(j in 1:k){
V[i,j] = Distance2(i,j)}}

## Evaluation de la distance de l'arbre détecté le plus proche pour chaque arbre inventorié
W = matrix(nrow=k,ncol=4)
for(i in 1:k){
W[i,1] = min(V[,i])
}

## Recherche de l'arbre qui est le plus proche.
Proche <- function(j){
i <- 1
while(V[i,j]>W[j,1]+0.1){
i <- i+1}
return(i)}

## Numéro de l'arbre détecté le plus proche pour tout arbre inventorié
for (j in 1:k){
W[j,2] = Proche(j)}

## Evaluation de la distance de l'arbre inventorié le plus proche pour chaque arbre détecté
W1 = matrix(nrow=max(N),ncol=2)
for(i in 1:max(N)){
W1[i,1] = min(V[i,])
}
## Recherche de l'arbre qui est le plus proche

```

```

Proche2 <- function(i){
j <- 1
while(V[i,j]>W1[i,1]+0.1){
j <- j+1}
return(j)}

## Numéro de l'arbre inventorié le plus proche pour tout arbre détecté
for (i in 1:max(N)){
W1[i,2] = Proche2(i)}

## Test de concordance "arbre détecté le plus proche pour arbre inventorié" VS "arbre inventorié le plus proche pour chaque arbre détecté"
for(j in 1:k){
W[j,3] = W1[W[j,2],2]
W[j,4] = ifelse((W[j,3]-j)>0,1,ifelse((W[j,3]-j)<0,1,0))
}

```

Annexe XI : cartes de la campagne de mesure en forêt de Bezange.

Remarque importante : pour l'ensemble des cartes qui suivent, les cercles concentriques gris représentent les distances par rapport au centre de la placette, à savoir 1, 5, 10, 15, 20 et 25 mètres.

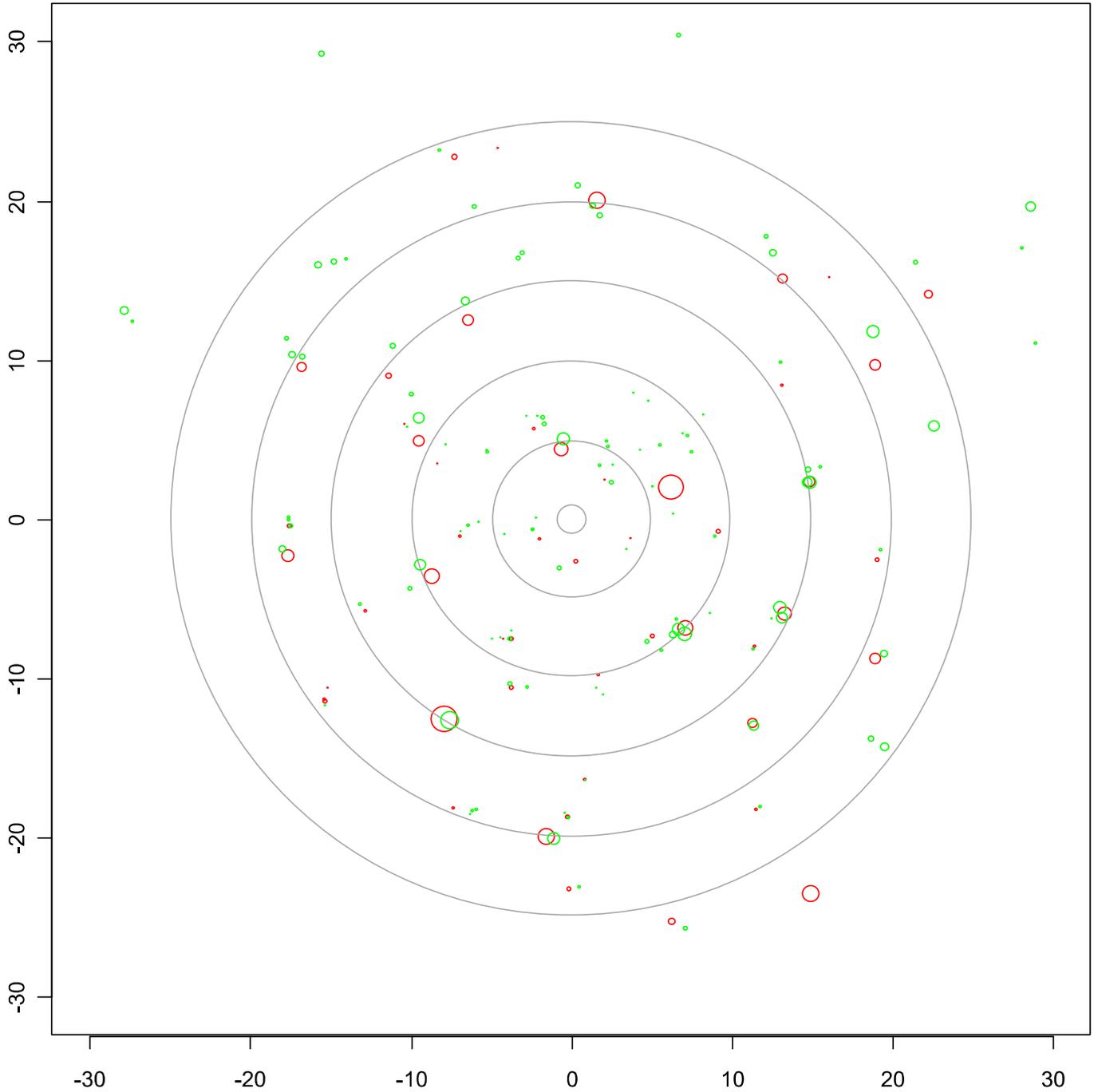


Figure 34: Carte des arbres inventoriés (rouge) et des arbres détectés (vert), placette 221

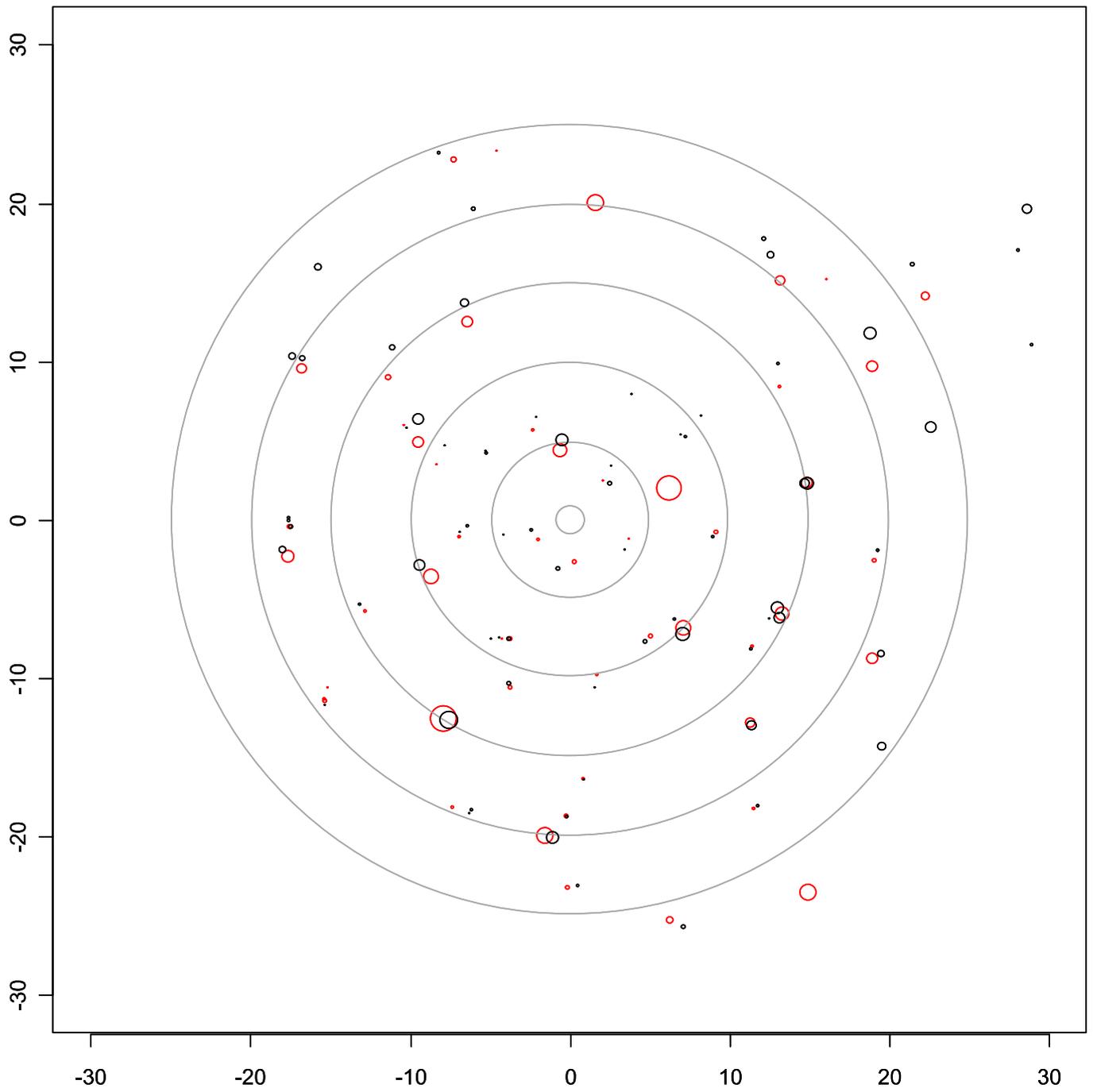


Figure 35: Carte des arbres inventoriés (rouge) et des arbres détectés (noir) après filtrage, placette 221

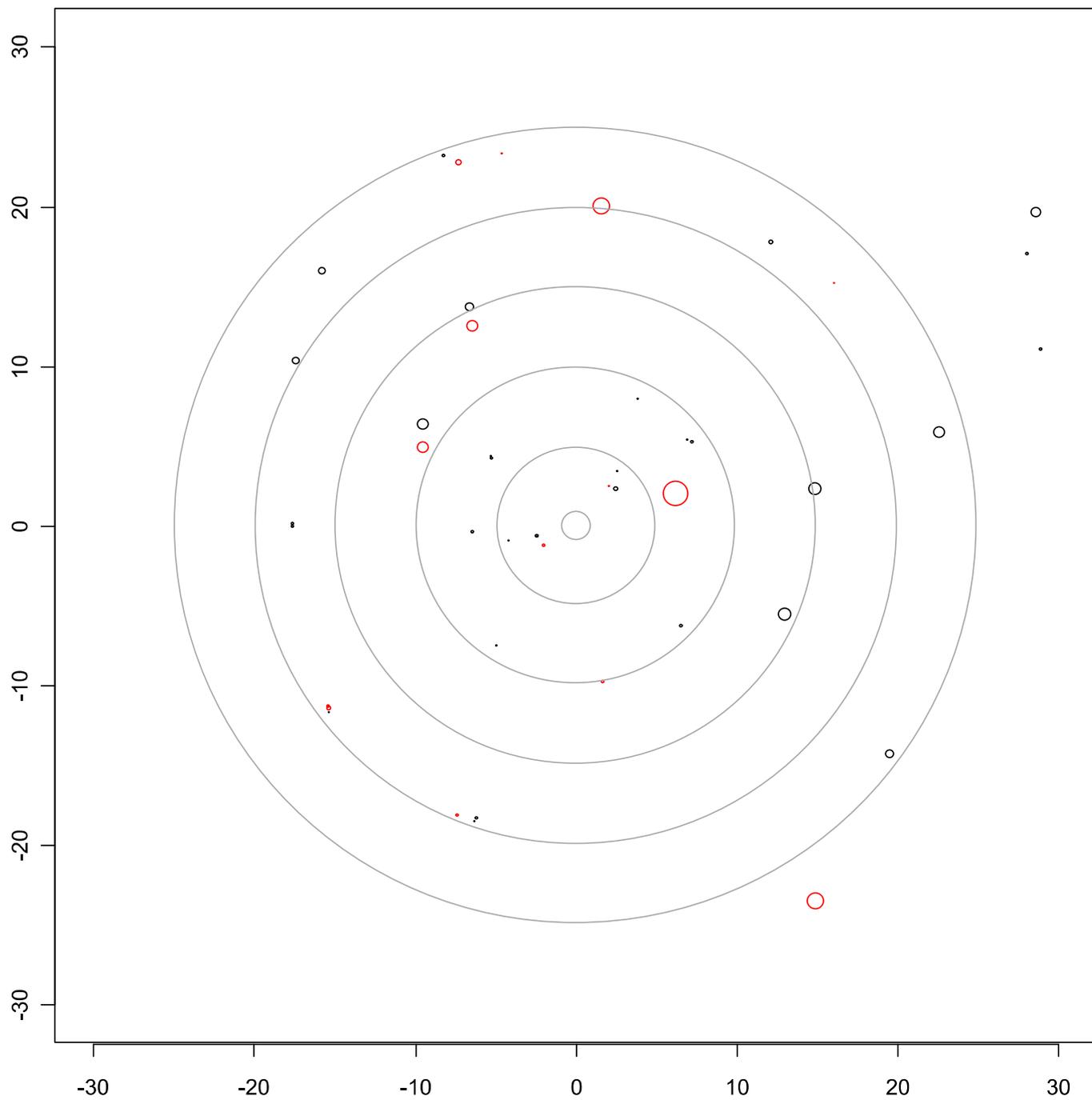


Figure 36: Carte des arbres inventoriés (rouge) et des arbres détectés (noir) non localisés automatiquement, placette 221

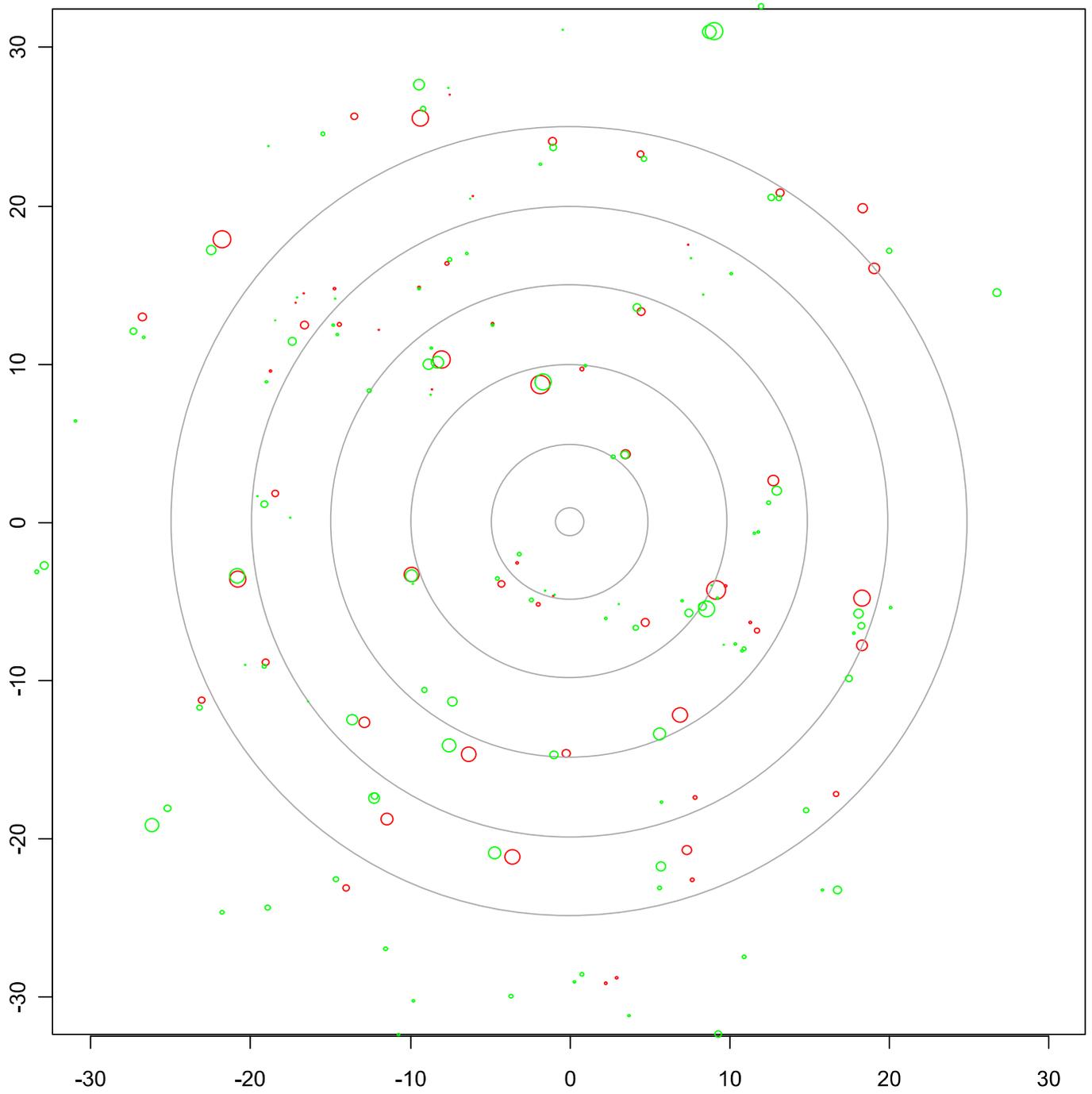


Figure 38: Carte des arbres inventoriés (rouge) et des arbres détectés (vert), placette 222

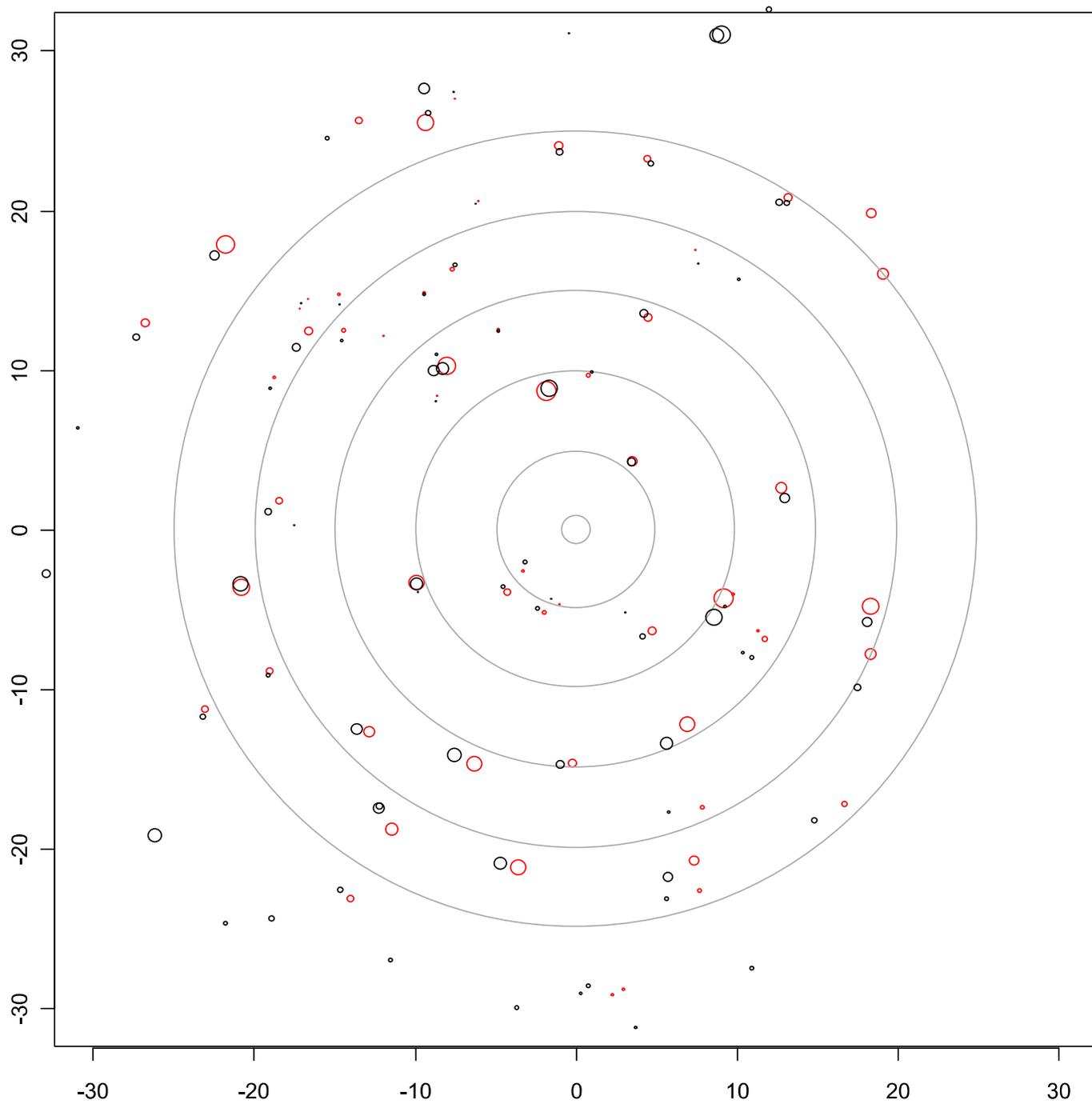


Figure 39: des arbres inventoriés (rouge) et des arbres détectés (noir) après filtrage, placette 222

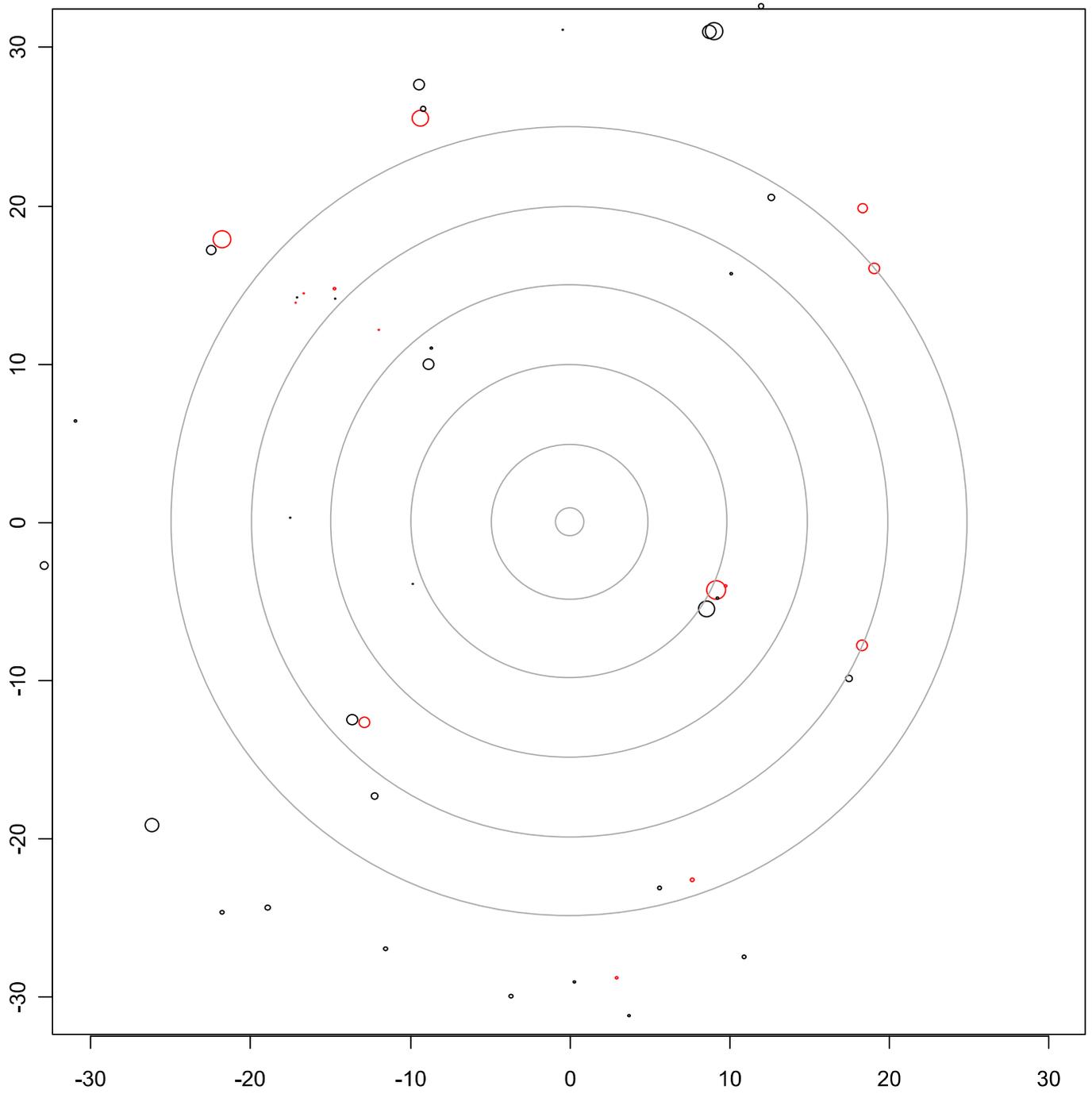


Figure 40: Carte des arbres inventoriés (rouge) et des arbres détectés (noir) non localisés automatiquement, placette 222

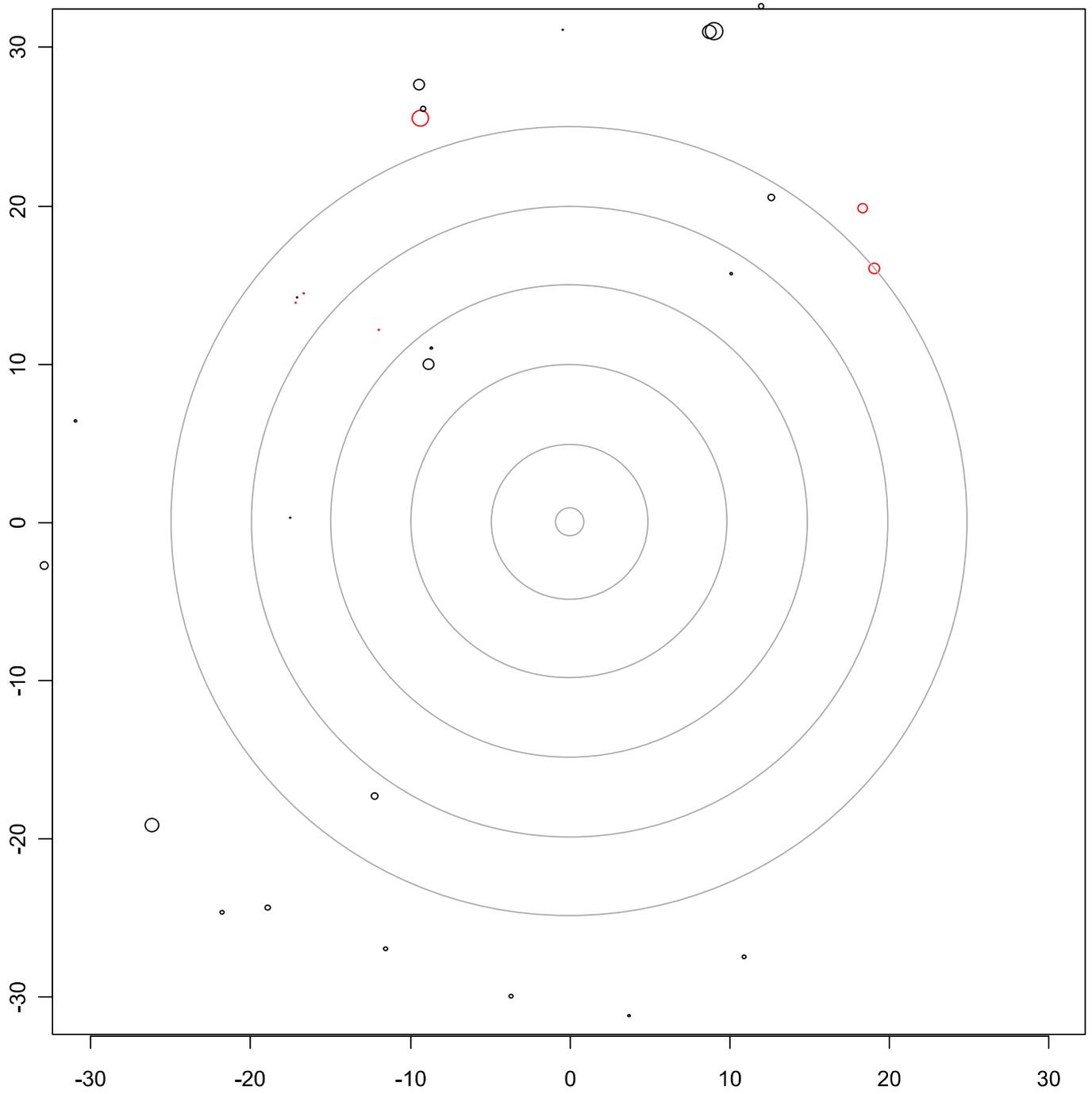


Figure 41: Carte des arbres inventoriés (rouge) et des arbres détectés (noir) après localisation manuelle, placette 222