

Fiche TD avec le logiciel : HouppierLIDAR

Estimation de la surface de houppiers de chênes sessiles

Jean R. LOBRY & Marie-Claude VENNER

La fiche explore différentes méthodes pour estimer la surface de houppiers : à partir de relevés sur le terrain, d'orthophotographies et de données LIDAR aériennes ou terrestres. On propose de visualiser la structure 3D de chênes sessiles et un exercice de détourage en 3D de houppier pour en estimer la surface.

Table des matières

1	Introduction	2
1.1	Les paquets  utilisés	2
1.2	Notes sur rgl	2
1.3	Les fichiers d'exemples	4
1.4	Géolocalisation des arbres d'intérêt	4
1.5	L'étalon d'argent	5
1.6	Limites des orthophotographies	6
2	Intérêt des données LIDAR	13
2.1	Les données LIDAR aériennes	13
2.2	Les données LIDAR au sol	18
3	Exercices de détourage 3D des houppiers	20
3.1	Détourage avec données LIDAR aériennes	20
3.2	Détourage avec données LIDAR terrestres	23
4	Annexes	28
4.1	Remerciements	28
4.2	Importation des données LIDAR aériennes	28
4.3	Importation des données LIDAR terrestres	29
4.4	Détourage des houppiers sur orthophotographies automnales	32
4.5	Zoom sur la canopée des houppiers d'intérêt	36
	Références	42

1 Introduction

LA surface du houppier d'un arbre est un *proxy* de la quantité d'énergie solaire auquel il a accès. C'est donc un paramètre biologique important pour caractériser un arbre.

1.1 Les paquets utilisés

POUR pouvoir faire cette fiche vous devez installer les paquets listés ci-après. Pour Ubuntu, on trouvera des indications pour installer le paquet *terra* dans une autre fiche¹.

```
library(rgl)
library(terra)
library(lidR)
library(concaveman)
```

1.2 Notes sur rgl

NOUS allons utiliser le paquet *rgl* [1] pour manipuler de façon interactive des nuages de points en 3D. Une démonstration valant mieux qu'un long discours, entrez le code ci-après puis manipulez avec le curseur de la souris le nuage de points dans la fenêtre qui est apparue.

```
chmin <- "https://pbil.univ-lyon1.fr/R/donnees/HouppierLIDAR/dfc.Rda"
load(url(chmin))
plot3d(dfc, size = 0.1, decorate = FALSE)
```

ON sera fort marri du résultat psychédétique obtenu (figure 1 page 3). Cet effet de moiré est dû au fait que pour le rendu 3D les calculs ne sont pas pris en charge par le processeur central mais par le processeur graphique. Les calculs de rendu 3D sont alors très rapides mais en simple précision, le processeur graphique est débordé par les grandes valeurs des latitudes et longitudes quand elles sont exprimées comme ici en LAMBERT 93. Pour pallier ce problème il suffit d'utiliser des valeurs plus raisonnables, par exemple en soustrayant la valeur minimale observée :

```
summary(dfc) # coordonnées en Lambert 93
      x           y           z
Min. :872720   Min. :6565720   Min. :316.7
1st Qu.:872725  1st Qu.:6565724  1st Qu.:317.6
Median :872730  Median :6565729  Median :337.2
Mean   :872730  Mean   :6565729  Mean   :332.8
3rd Qu.:872736  3rd Qu.:6565734  3rd Qu.:344.6
Max.   :872741  Max.   :6565740  Max.   :349.1
dfcn <- as.data.frame(lapply(dfc, \((x) x - min(x)))
```

summary(dfcn) # coordonnées translatées (m)

	x	y	z
Min.	0.000	0.000	0.0000
1st Qu.	4.783	4.568	0.9184
Median	9.733	9.409	20.5214
Mean	10.043	9.432	16.1048
3rd Qu.	15.424	14.420	27.8529
Max.	20.326	19.620	32.4361

1. <https://pbil.univ-lyon1.fr/R/pdf/PointsGPS.pdf>

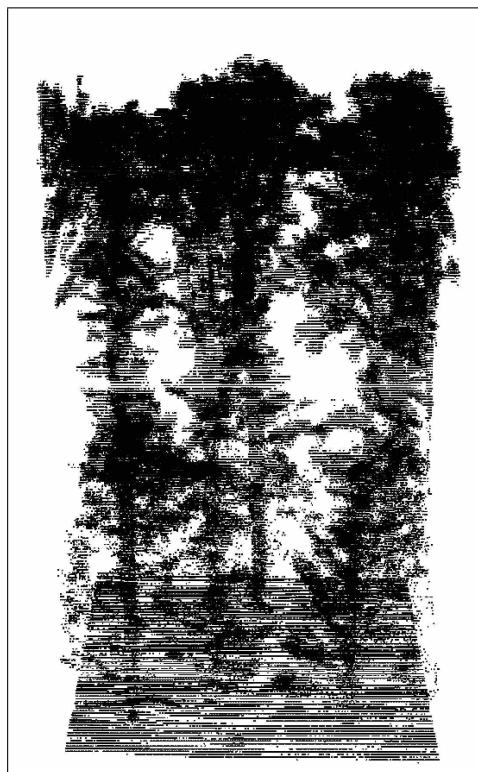


FIGURE 1 – Exemple de résultat obtenu avec la fonction `plot3d()` du paquet `rgl` quand on visualise le nuage de points avec ses coordonnées originelle en LAMBERT 93.

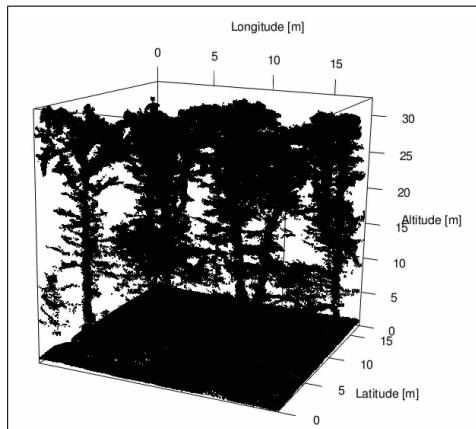


FIGURE 2 – Exemple de résultat obtenu avec la fonction `plot3d()` du paquet `rgl` quand on visualise le nuage de points avec des coordonnées translatées.

UTILISEZ le code ci-après pour jouer avec le nuage de points dans ce nouveau système de coordonnées, le résultat est bien plus satisfaisant (figure 2 page 4). Nous avons plus d'un million de points à visualiser ici, les performances de la carte graphique sont assez époustouflantes.

```
plot3d(dfcn, size = 0.1, xlab = "Longitude [m]",
       ylab = "Latitude [m]", zlab = "Altitude [m]",
       zlim = c(2, 35))
```

1.3 Les fichiers d'exemples

LES fichiers d'exemples de cette fiche de TD sont distribués sur le serveur pédagogique². Le plus simple est de recopier l'ensemble du dossier `HouppierLIDAR` dans son dossier de travail pour reproduire les résultats plutôt que de télécharger les fichiers à la volée comme dans la section 1.2 page 2.

1.4 Géolocalisation des arbres d'intérêt

ON ne cherche pas ici à faire une étude de la surface des houppiers d'une population d'arbres dont peu nous chaut la localisation exacte, mais à estimer, le plus précisément possible, cette statistique pour des arbres *d'intérêt*. Un prérequis est donc d'avoir une bonne géolocalisation d'iceux, disons avec une résolution de l'ordre du mètre (table 1 page 5) puisque que les surfaces des houppiers qui nous intéressent font de l'ordre de 100 m² (soit approximativement un cercle de 6 m de rayon). Le nom des arbres est le suivant :

```
allTrees3char <- c("103", "106", "109", "110", "114", "301", "302", "303", "304", "305")
(allTrees <- paste0("CHS01_", allTrees3char))
```

2. <https://pbil.univ-lyon1.fr/R/donnees/HouppierLIDAR/>

[1] "CHS01_103" "CHS01_106" "CHS01_109" "CHS01_110" "CHS01_114" "CHS01_301"
[7] "CHS01_302" "CHS01_303" "CHS01_304" "CHS01_305"

DANS la version longue le nom des arbres commence par CHS01 parce qu'ils sont dans le pourtour immédiat de la placette RENECAFOR³ de code CHS01, soit des chênes sessiles dans le département de l'AIN⁴.

Arbre	Longitude [°]	Latitude [°]	Altitude [m]	Houppier
103	5.23830713	46.16930104	316.635	112.9
106	5.23906252	46.16937754	314.258	70.9
109	5.23894412	46.17033698	315.181	78.3
110	5.23893525	46.17043837	314.949	94.2
114	5.23823214	46.17042410	320.943	88.0
301	5.23855780	46.17067747	315.128	90.7
302	5.23882226	46.17058762	314.741	58.9
303	5.23799744	46.17043303	320.040	146.9
304	5.23903503	46.16959421	315.737	25.5
305	5.23910111	46.16915981	313.902	147.3

TABLE 1 – Les coordonnées GPS (EPSG:4326) des arbres suivis. Données relevées sur le terrain le 2022-07-04 par François DÉBIAS avec la correction RTK (Real Time Kinematic du réseau centipède) et une précision minimale de 25 cm pour la latitude et la longitude annoncée par le fournisseur de correction différentielle. Cependant, en milieux forestier, les problématiques multitrajets liées aux branches et troncs dégradent fortement la précision réelle du positionnement et on ne peut espérer au mieux qu'une précision de l'ordre de 1 à 2 m (Laurent MALABEUX, communication personnelle). La surface des houppiers en m^2 est celle déterminée sur le terrain (section 1.5 page 5). Pour la visualisation de ces données voir la figure 4 page 7.

1.5 L'étalon d'argent

UNE façon simple et « économique » de déterminer la surface des houppiers d'arbres d'intérêt consiste à se déplacer sur le terrain et à mesurer les distances entre le tronc et l'extrémité du houppier selon quatre directions orthogonales (figure 3 page 6), par exemple en suivant les quatre points cardinaux (d_N , d_S , d_E et d_O). On approxime ensuite le houppier par une ellipse dont on peut calculer la surface, S , ainsi :

$$S = \pi \frac{1}{2} (d_N + d_S) \frac{1}{2} (d_E + d_O) \quad (1)$$

3. Sigle [3, 12, 7] pour le « Réseau National de suivi à long terme des ÉCOsystèmes FORestiers ». Comme le conseille J. ANDRÉ [2], on respectera ici le choix des auteurs de ne pas utiliser de capitales accentuées, tout en se souvenant que cela se prononce RÉNÉCAFOR.

4. Pour être plus précis, allant du plus général au plus particulier, les arbres d'intérêt sont dans l'hémisphère nord, en EUROPE continentale, en FRANCE métropolitaine, dans le département de l'AIN, dans la région de la DOMBES, dans la commune de PÉRONNAS, dans la parcelle cadastrale section C n° 206 p au lieu-dit « les COUPES BLANCHES » [4, p. 40], dans la parcelle forestière C3 de la forêt domaniale de SEILLON [4, p. 17], dans le pourtour immédiat de la placette RENECAFOR CHS01, aux coordonnées GPS données dans la table 1 page 5.

CHS01 103	47,27	52,66	5,5	7	7,5	4
CHS01 106	46,64	0	3,5	6	5,5	4
CHS01 109	50,88	55,40	5,5	5	5,5	4
CHS01 110	45,24	0	5,5	6,5	6	4
CHS01 114	47,36	54,96	7	7	5,5	2,5
CHS01 301	52,55	58,67	6,5	4,5	6,5	4
CHS01 302*	42,2	45,34	7,5	2,5	3	4,5
CHS01 303	60,51	62,29	5	6	9,5	7,5
CHS01 304	38,5	40,11	2,5	2,5	5	1,5
CHS01 305	68,94	76,28	8	7	6	6,5

FIGURE 3 – Exemple de relevé fait sur le terrain en forêt de SEILLON le 2023-08-18. Les quatre dernières colonnes donnent pour chaque arbre les distances en mètre entre le tronc et l’extrémité du houppier selon quatre directions orthogonales (nord, sud, est, ouest).

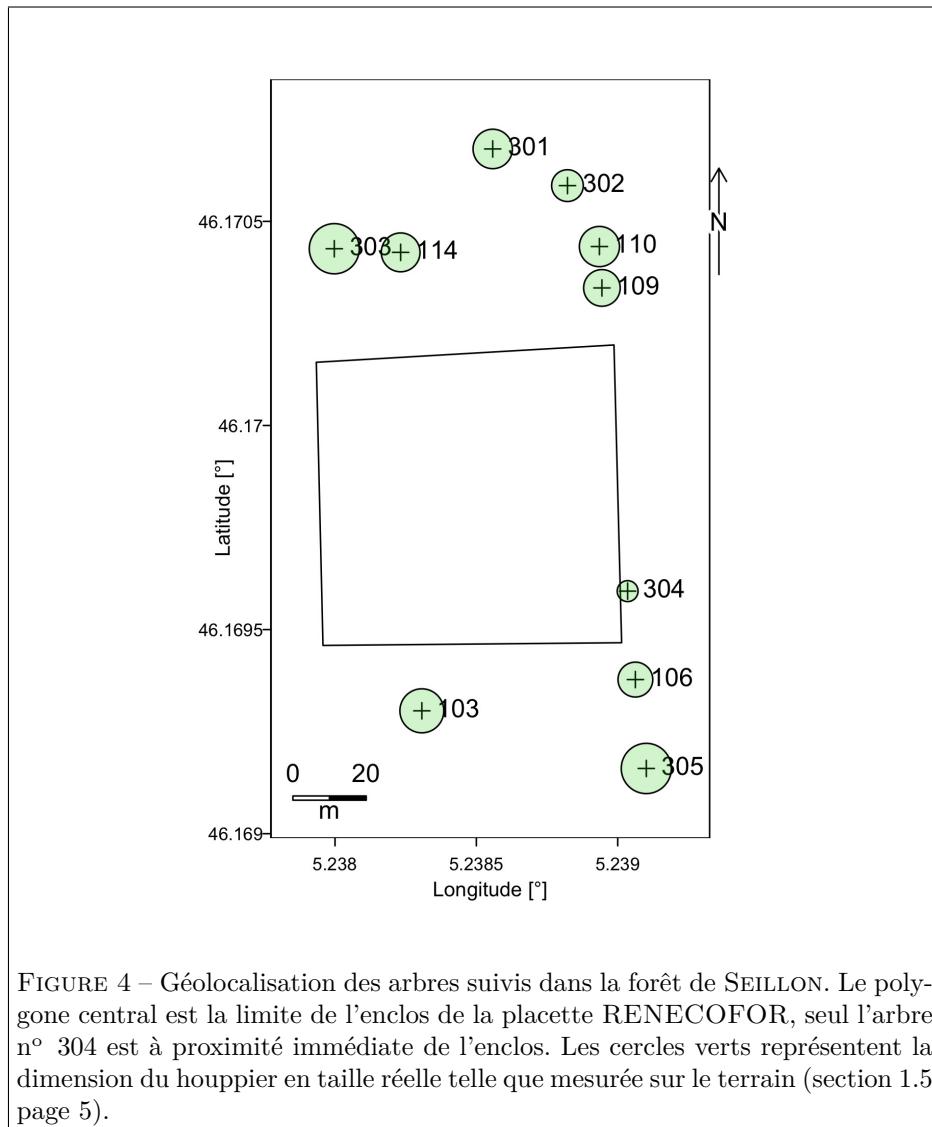
CETTE méthode ne peut donner que des résultats approximatifs (table 2 page 12) mais est extrêmement *robuste* dans le sens où, sauf erreur clé-ricale, elle ne peut conduire ni à des estimations radicalement inexactes ni à de mauvaises identifications. On ne peut pas parler d’étalon or ici mais le qualificatif d’étalon d’argent ne nous semble pas usurpé. Toute estimation de la surface des houppiers par une méthode alternative devra être examinée à l’aune de cet étalon.

CETTE méthode de « terrain » pour la détermination de la surface des houppiers n’est pas une panacée et comporte ses limites. D’un point de vue économique elle ne vaut que pour des arbres d’intérêt en effectif restreint et il serait trop dispendieux de la déployer à l’échelle d’une forêt entière (voyez la figure 5 page 9 pour vous représenter l’ampleur de la tâche que cela nécessiterait). D’un point de vue pratique encore faut-il que le terrain soit accessible : que faire si un bosquet de houx impénétrable ou une doline vient empêcher de se déplacer *in situ* en un point nécessaire ? Ce n’est pas le cas ici en forêt de SEILLON sauf peut-être de façon anecdotique pour l’arbre n° 304 où la clôture de la parcelle RENECOFOR empêche de se déplacer facilement à l’ouest (figure 4 page 7).

1.6 Limites des orthophotographies

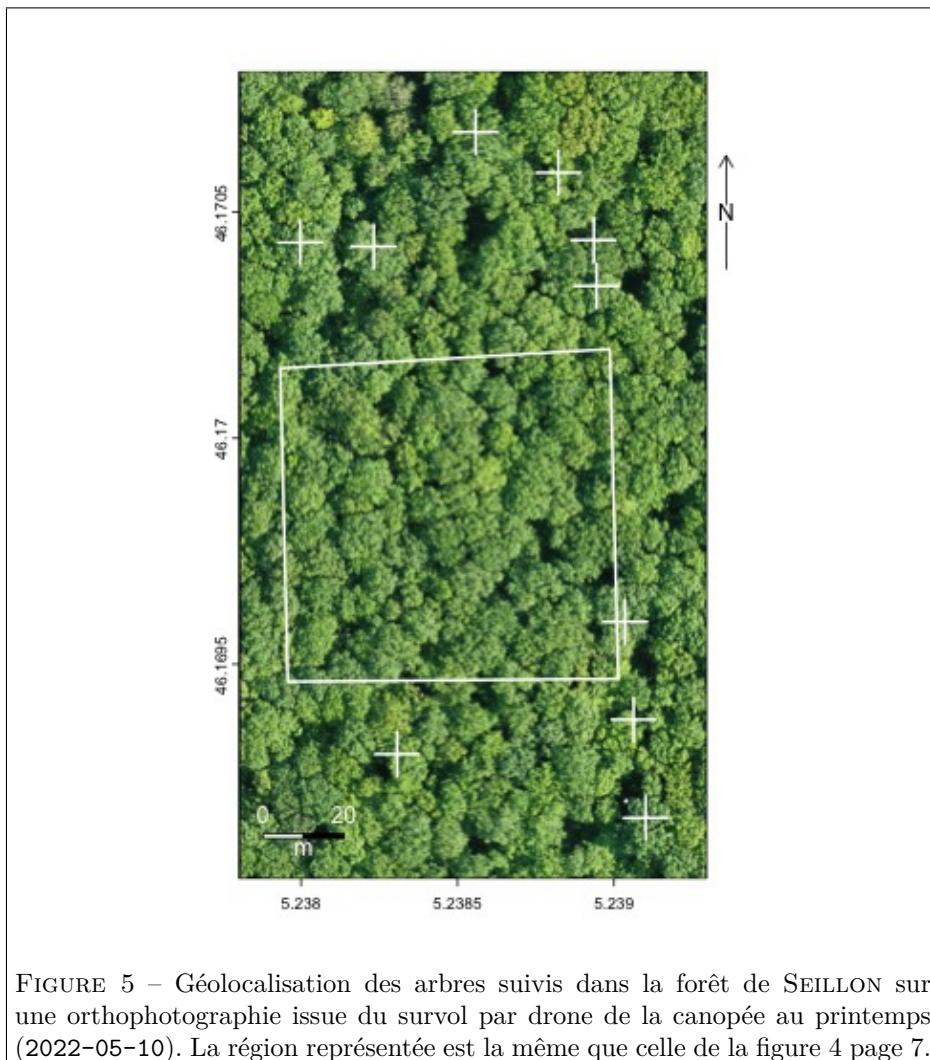
LES orthophotographies sont obtenues en faisant voler un drone au dessus de la canopée pour prendre des clichés qui sont traités pour produire une image vue en tout point comme étant à la verticale du sol. Le drone possédant un GPS et une centrale inertIELLE, ces points sont géolocalisés.

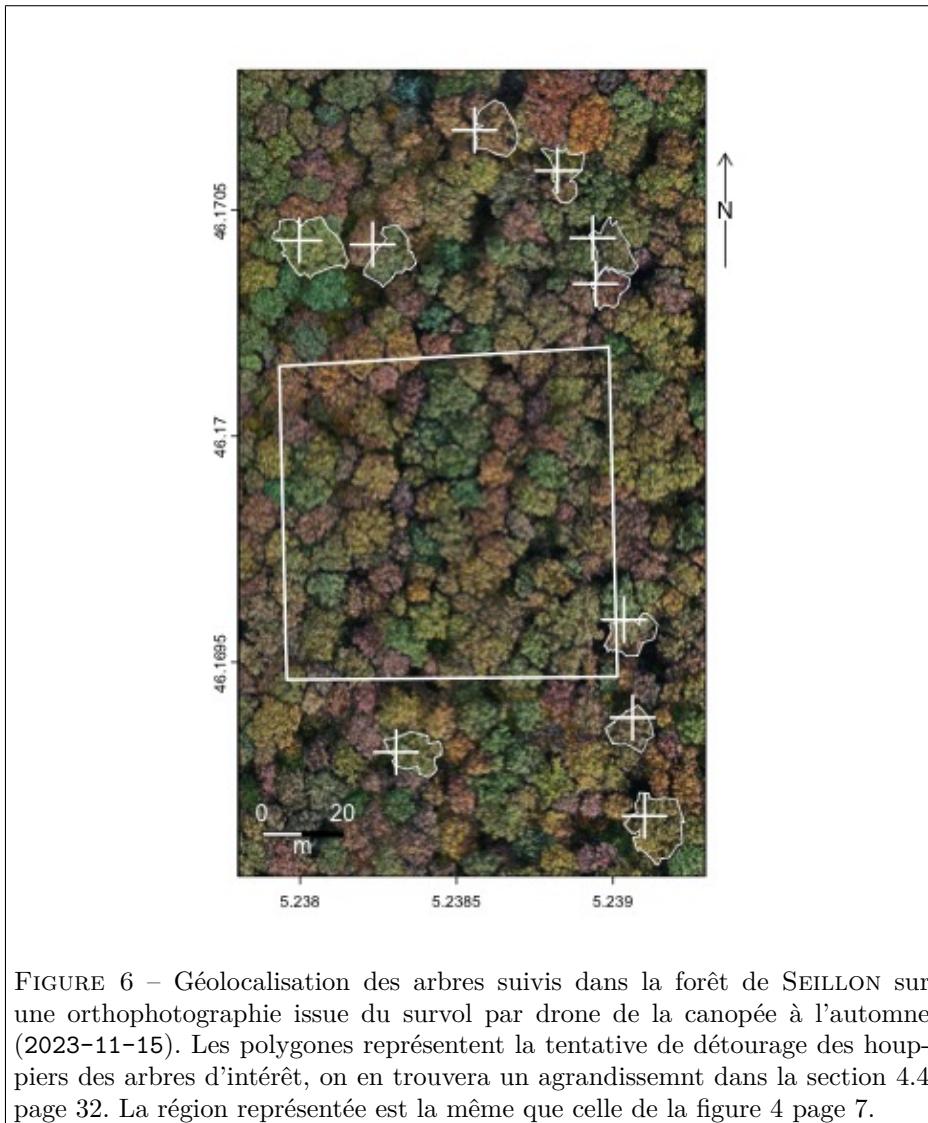
LE survol de la canopée au printemps (figure 5 page 9) illustre les deux problèmes auxquels nous sommes confrontés pour estimer la surface des houp-

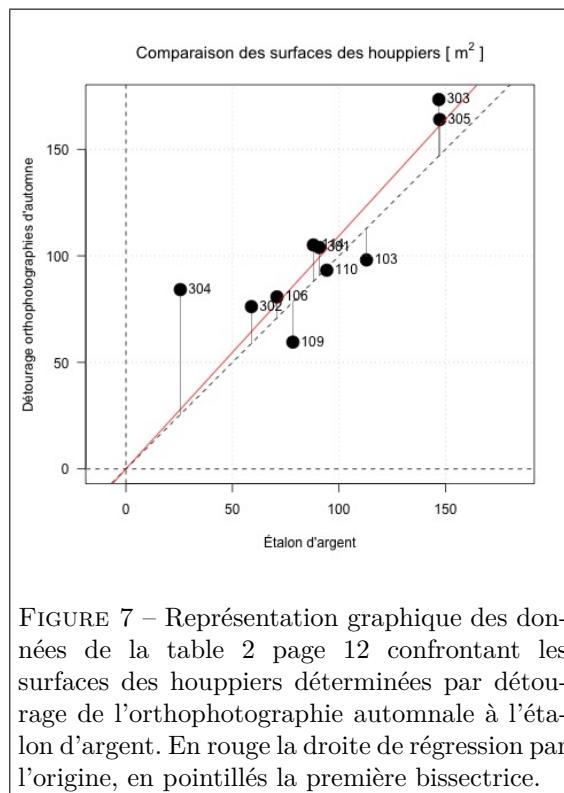


piers à partir de ces données. Premièrement, la délimitation précise des houppiers n'est pas toujours évidente parce qu'il y a peu de contraste de couleur entre eux : c'est le printemps, tous les arbres sont verts. Secondement, l'identification du bon houppier d'un arbre d'intérêt n'est pas certaine à cause de l'imprécision de la localisation des points du drone.

UNE façon « naturelle » d'augmenter le contraste entre les houppiers est de survoler la canopée en automne (figure 6 page 10) pour tirer parti des différences dans la phénologie de la sénescence foliaire : les feuilles ne jaunissent pas au même moment d'un individu à l'autre. On pourrait se croire tiré d'affaire avec une image aussi contrastée, mais ce n'est pas aussi simple : d'une part, la phénologie de deux houppiers voisins peut être similaire conduisant à un regroupement erroné, et, d'autre part, la phénologie n'est pas forcément identique au sein d'un même individu, une mutation somatique ayant très bien pu se propager tout au long d'une branche.







La figure 6 page 10 illustre le détourage des houppiers effectué à partir de l'orthophotographie issue du survol par drone de la canopée en automne. Le centre des houppiers n'est pas nécessairement à la verticale des troncs, mais c'est ici une approximation raisonnable parce que le terrain est très homogène et que les arbres sont cultivés pour produire de longs troncs rectilignes. Notons que les arbres d'intérêt ne sont pas voisins de l'orée, on n'a donc pas à craindre un effet lisière de phototropisme poussant latéralement les houppiers vers plus de lumière. Sous cette approximation raisonnable, les coordonnées de l'orthophotographie de l'arrière-saison semblent translatées de quelques mètres vers le sud et l'est (figure 6 page 10) par rapport aux coordonnées de référence des troncs (table 1 page 5). Ceci illustre la relative imprécision des coordonnées GPS, imprécision qui nous empêche d'être absolument certains de ne pas avoir fait d'erreur d'identification des houppiers.

La figure 7 page 11 confronte la surface des houppiers obtenue par détourage de l'orthophotographie automnale avec l'éta-
lon d'argent. Les deux méthodes donnent des valeurs commensurables, de l'ordre de 100 m² pour la surface des houppiers, la pente de la droite de régression par l'origine ($r^2 = 0.96$) est de 1.1, soit très proche de celle de la première bissectrice, mais nous sommes loin d'un alignement parfait des planètes. La médiane des valeurs absolues des erreurs relatives est de l'ordre de 15 %, ce qui n'est pas d'une précision transcen-
dantale, mais pourrait être satisfaisant s'il n'y avait pas le cas de l'arbre l'arbre n° 304 avec un excès de +59 m² pour une valeur de référence de 25.5 m², soit

Arbre	Ref	Det	Écart abs.	Écart rel.
110	94.25	93.245	-1.003	-1.064
305	147.26	163.970	16.708	11.346
103	112.90	98.093	-14.808	-13.116
106	70.88	80.739	9.857	13.906
301	90.71	103.920	13.207	14.558
303	146.87	173.350	26.481	18.030
114	87.96	105.100	17.135	19.480
109	78.34	59.502	-18.841	-24.050
302	58.90	76.168	17.263	29.307
304	25.53	84.151	58.626	229.675

TABLE 2 – Les surfaces des houppiers en m^2 avec dans l'ordre la valeur de référence, celle déterminée par détournage de l'orthophotographie automnale, l'écart entre les deux en absolu et relatif. Pour la visualisation de ces données voir la figure 7 page 11.

une erreur relative de +230 %, plus du triple de la valeur de l'étalon d'argent ! L'examen de la figure 26 page 35 ne permet pas de mettre en évidence d'erreur cléricale dans le détournage du houppier de l'arbre n° 304 sur l'orthophotographie automnale. Mais nous sommes ici un petit peu dans nos petits souliers parce l'arbre n° 304 était précisément celui pour lequel l'étalon d'argent n'était pas des plus massif à cause d'une difficulté sur le terrain pour déterminer la distance du houppier au tronc vers l'ouest. Pour lever les ambiguïtés il va très clairement nous falloir accéder à la troisième dimension.

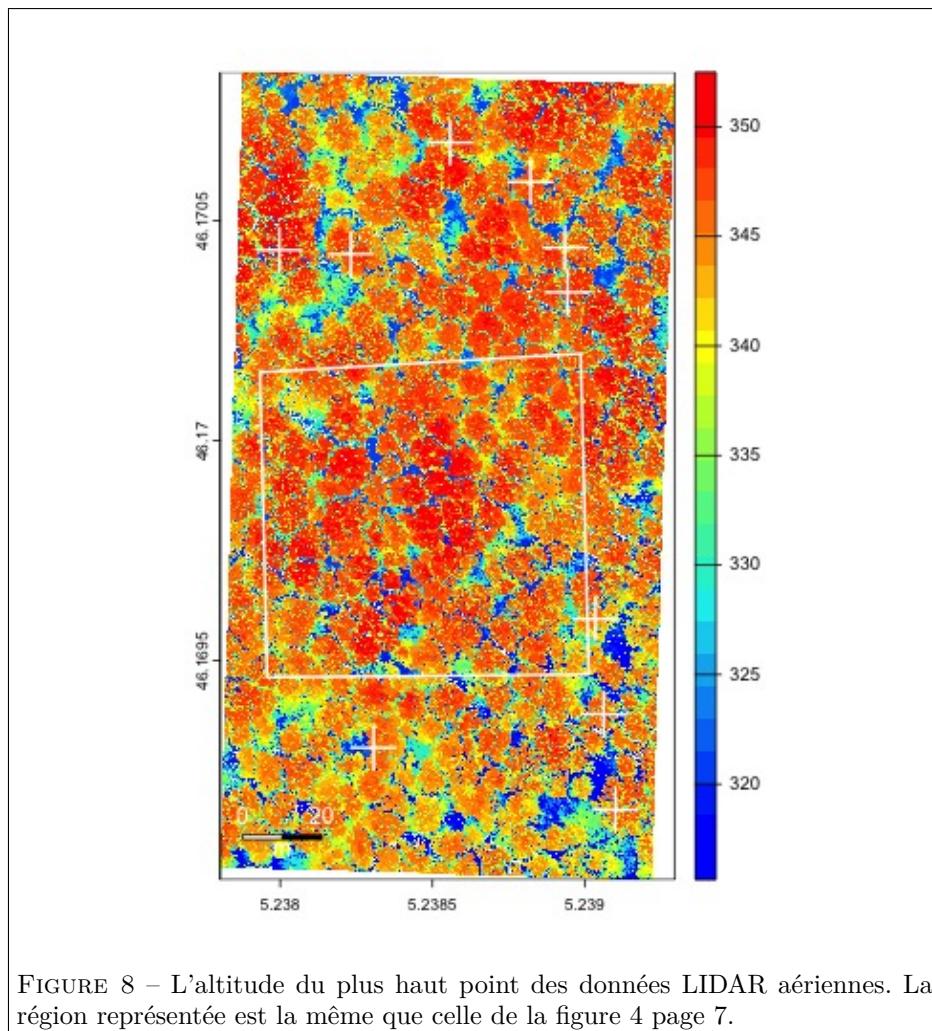
2 Intérêt des données LIDAR

LES données LIDAR sont des coordonnées spatiales (x, y, z) de points appartenant à la surface des arbres obtenues en analysant les retours d'impulsions laser. Elles permettent donc appréhender la structure en trois dimensions des arbres. On peut équiper un drone d'un LIDAR et dans ce cas on aura une très bonne vision de la surface de la canopée, mais avec des zones d'occlusion possibles au niveau du sol. On peut également faire des relevés LIDAR au sol, avec une très bonne vue de la structure du houppier vue du dessous, mais avec des zones d'occlusion possibles au sommet des houppiers. Nous allons combiner ces deux approches pour estimer au mieux la surface des houppiers des arbres d'intérêt.

2.1 Les données LIDAR aériennes

LE fichier des données originelles est trop volumineux pour être mis à disposition des étudiants sur le serveur pédagogique, mais on trouvera à titre informatif le code utilisé en annexe (section 4.2 page 28). Des données dérivées moins volumineuses sont néanmoins disponibles. Par exemple, le code suivant permet de reproduire la figure 8 page 14. C'est une image des plus hauts points détectés par le LIDAR, c'est à dire la surface de la canopée quand on est en forêt. On ajoute ici une croix blanche pour la position de référence de l'arbre (table 1 page 5), l'enclos de la parcelle RENECAFOR et une barre d'échelle. Le sol apparaît ici en bleu, à une altitude de 315 m, ce qui est cohérent avec les données de référence pour les arbres (table 1 page 5). La canopée culmine à 350 m, soit à 35 m au dessus du sol, ce qui est cohérent avec la taille des arbres, mais pour être plus précis on a besoin d'un modèle de l'altitude du sol.

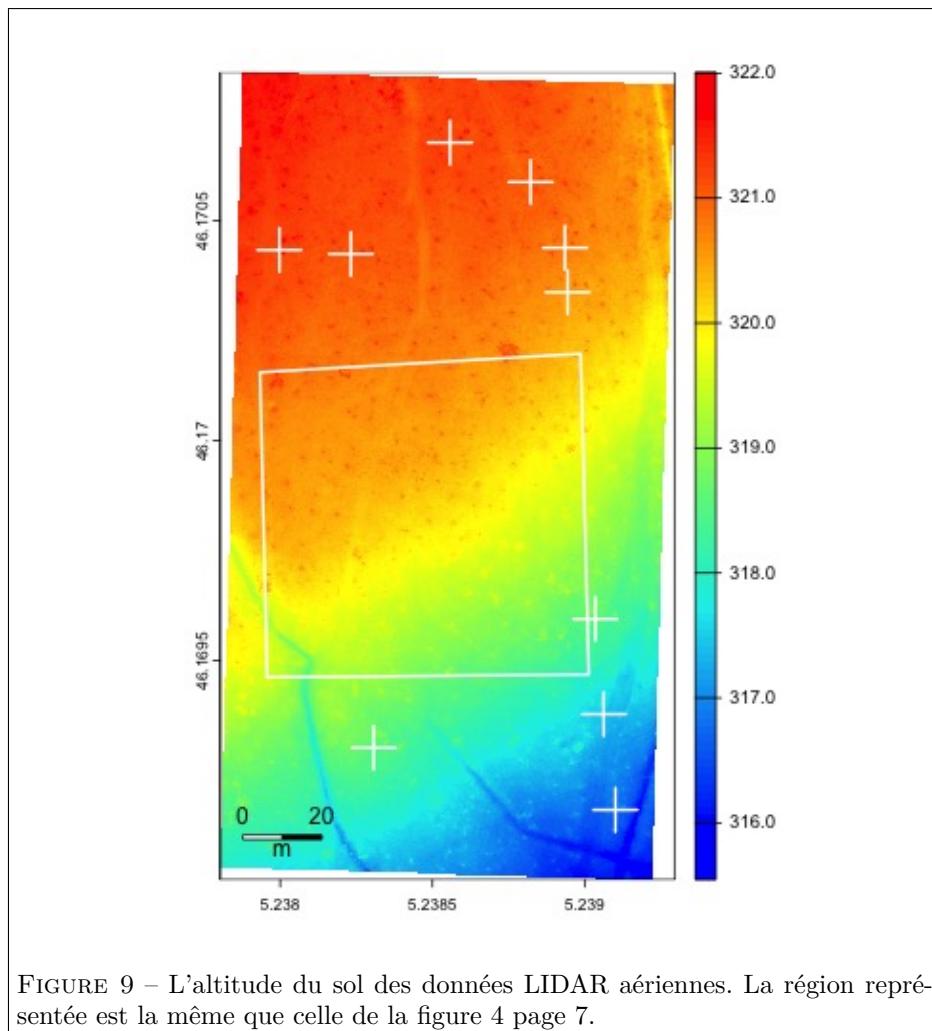
```
dsmGPS <- rast("HouppierLIDAR/dsmGPS.tif")
plot(dsmGPS, col = lidR::height.colors(25))
mydeco <- function(){
  centers <- vect("HouppierLIDAR/centers")
  points(centers, cex = 3, pch = 3, col = "white", lwd = 2)
  enclos <- vect("HouppierLIDAR/enclos")
  polys(enclos, col = "transparent", border = "white", lwd = 2)
  sbar(d = 0.02, type = "bar", below = "m", scaleby = 0.001, divs = 2, col = "white")
}
mydeco()
```

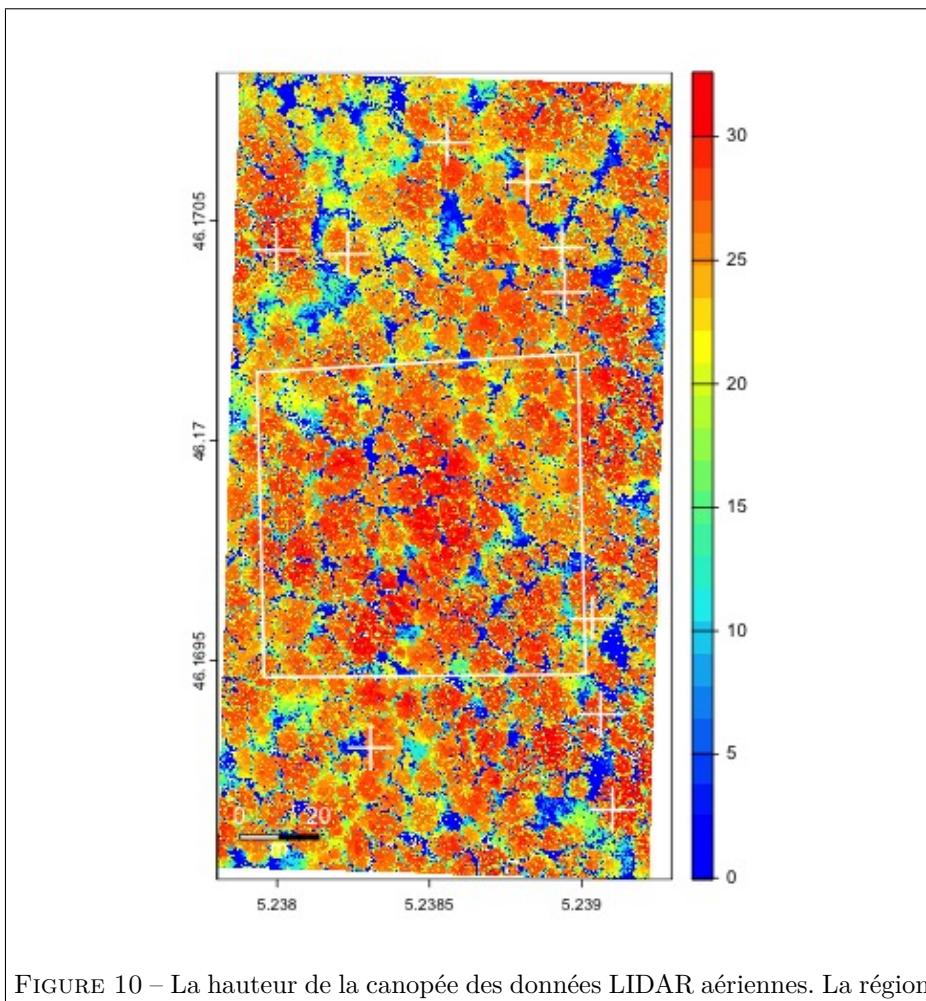


Le code suivant permet de reproduire le modèle du sol⁵ de la figure 9 page 16. Le terrain est très peu accidenté avec un dénivélé de 6 m entre le nord-ouest et le sud-est de la zone d'intérêt, soit une pente de moins de 3 %. Pour qui connaît le terrain on reconnaît bien le petit ruisseau qui coupe le coin sud-ouest de la placette RENECAFOR, ce qui signifie qu'il n'y a pas d'erreur grossière dans la position des points issus du drone-LIDAR. Les données LIDAR aériennes permettent donc de faire virtuellement une opération de coupe blanche en rasant la canopée pour mettre en évidence des structures au niveau de la topologie du sol, ce qui peut être assez spectaculaire pour les recherches archéologiques, mais ce n'est pas notre objet ici. On comprend bien en revanche que si on s'intéresse à la hauteur des arbres il est indispensable de corriger par la hauteur du sol sans quoi les arbres de la zone nord feront 6 m de plus que ceux de la zone sud.

```
dtmGPS <- rast("HouppierLIDAR/dtmGPS.tif")
plot(dtmGPS, col = lidR::height.colors(100))
mydeco()
```

5. Pour détecter les points du sol j'ai utilisé la fonction `lidR::classify_ground()` avec la méthode CSF (*Cloth Simulation Filter*) du paquet `RCSF` [10]. Elle revient à retourner le nuage de points et à poser dessus à l'envers un filet de pêche plus ou moins rigide aux mailles plus ou moins grandes et à laisser lui prendre sa position d'équilibre. Les points suffisamment proches du filet (50 cm par défaut) sont considérés comme étant au sol. La forêt de SEILLON étant très plate j'ai utilisé un paramètre de rigidité élevé (`rigidness = 2`) avec des mailles larges (`cloth_resolution = 1`).





Le code suivant permet de produire le modèle de hauteur de la canopée de la figure 10 page 17. Par rapport à l'orthophotographie automnale (figure 6 page 10) on peut constater que les houppiers sont bien mieux résolus ce qui facilitera grandement le détournage d'iceux. On trouvera en annexe (section 4.5 page 36) des agrandissements de ce modèle de hauteur de la canopée centrés sur les arbres d'intérêt. Tout n'est pas encore gagné, nous avons toujours un problème d'identification, par exemple pour l'arbre n° 114 (figure 32 page 38), à cause de la relative imprécision de la localisation des points. C'est ici que les données LIDAR au sol vont nous être utiles pour lever toute ambiguïté.

```
chmGPS <- dsmGPS - dtmGPS
plot(chmGPS, col = lidR::height.colors(100))
mydeco()
```

2.2 Les données LIDAR au sol

LES données originelles sont encore ici trop volumineuses pour être mises à disposition des étudiants sur le serveur pédagogique, on trouvera pour information en annexe (section 4.3 page 29) les pré-traitements effectués. Les dossiers `zoom_*` contiennent les données au format *shapefile* d'un agrandissement autour des arbres d'intérêt. Utilisez le code suivant pour visualiser l'arbre n° 303, puis faites tourner le nuage de point pour le regarder par en-dessous comme dans la figure 11 page 19. Les données LIDAR au sol nous permettent de lever toute ambiguïté pour l'identification des houppiers grâce à la présence des repères sur le terrain que constituent les dispositifs de collecteurs de glands dont sont équipés les arbres d'intérêt.

```
plotZoom <- function(the_tree, ncol = 100,
                      xplt = c(0, 100), yplt = c(0, 100), zplt = NULL, maxpt = 10^6){
  zoom <- vect(paste0("HouppierLIDAR/zoom_", the_tree))
  x <- geom(zoom)[,"x"] ; x <- 100*(x - min(x))/(max(x) - min(x))
  y <- geom(zoom)[,"y"] ; y <- 100*(y - min(y))/(max(y) - min(y))
  z <- zoom$z
  df <- data.frame(x = x, y = y, z = z)
  zfac <- cut(z, breaks = ncol)
  df$col <- lidR::height.colors(ncol)[zfac]
  df <- df[df$x > xplt[1] & df$x < xplt[2], ]
  df <- df[df$y > yplt[1] & df$y < yplt[2], ]
  if(!is.null(zplt)) df <- df[df$z > zplt[1] & df$z < zplt[2], ]
  df <- df[sample(1:nrow(df), min(maxpt, nrow(df))), ]
  plot3d(df, size = 0.1, col = df$col, zlim = range(z),
         xlim = c(0, 100), ylim = c(0, 100))
}
plotZoom("CHS01_303")
```

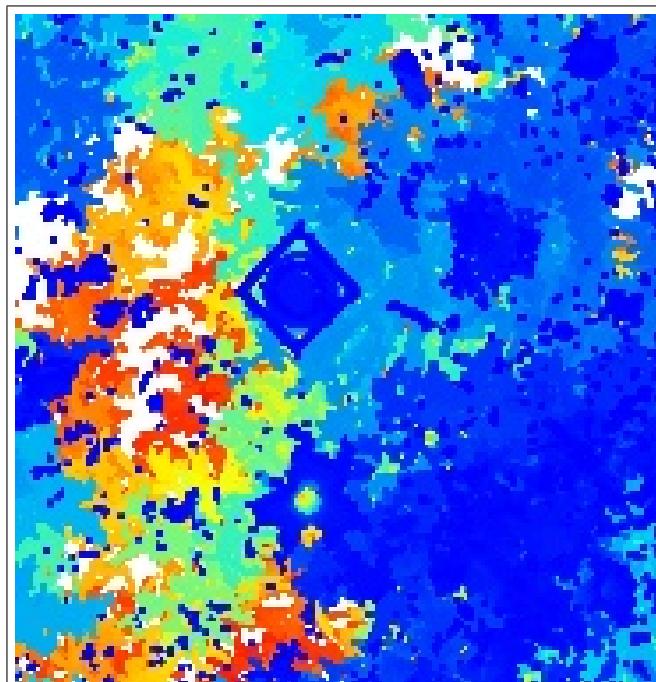


FIGURE 11 – Vue par en-dessous des données LIDAR terrestres au voisinage de l'arbre n° 303. On distingue très bien le collecteur de glands à proximité du tronc : c'est un récipient cylindrique protégé de la curiosité des sangliers et autres amateurs de glands par un petit enclos carré. On est donc certain ici de la bonne identification pour l'arbre n° 303.

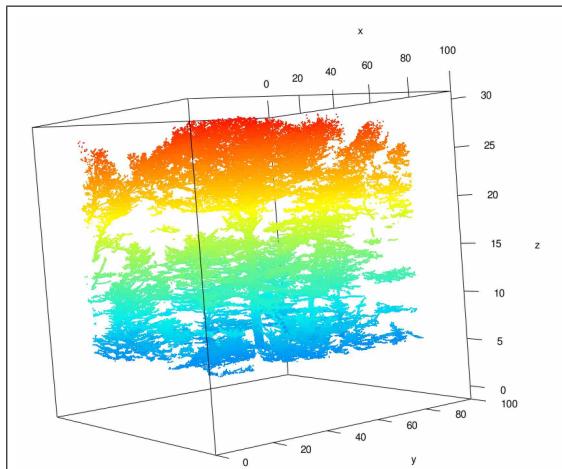


FIGURE 12 – Une vue des données LIDAR terrestres au voisinage du houppier de l’arbre n° 303. On voit ici comment il oblige son voisin à se plier pour accéder à la lumière.

LES arguments [x|y|z]plt de la fonction `plotZoom()` permettent de censurer en marge des points tout en conservant l’échelle originelle pour se rapprocher au plus près de l’arbre d’intérêt. L’argument `maxpt` contrôle le nombre maximum de points à afficher pour alléger l’image. Essayez le code suivant pour reproduire la figure 12 page 20. Notez comme les détails du houppier sont visibles de façon extrêmement détaillée. Grâce aux données du LIDAR terrestre nous n’aurons plus de problème d’identification des houppiers.

```
plotZoom("CHS01_303", xplt = c(10, 90), yplt = c(10, 90), zplt = c(5, 30), maxpt = 5*10^5)
```

3 Exercices de détourage 3D des houppiers

NOUS vous proposons comme exercices d’essayer de détourer le houppier de l’arbre n° 305 parce qu’il est des plus majestueux. Nous essayerons à partir des données LIDAR aériennes et terrestres.

3.1 Détourage avec données LIDAR aériennes

COMMENCEZ par récupérer le nuage de points au voisinage de l’arbre n° 305, il y a environ un million de points sur une surface de 400 m² et les coordonnées sont en LAMBERT 93.

```
load("HouppierLIDAR/pcloc_CHS01_305.Rda")
print(pcloc)
  class       : LAS (v1.2 format 3)
  memory     : 33.6 Mb
  extent      : 872724.8, 872744.6, 6565698, 6565718 (xmin, xmax, ymin, ymax)
  coord. ref. : RGF93 v1 / Lambert-93
  area        : 396 m2
```

```
points      : 1.26 million points
density     : 3180.88 points/m2
```

POUR ne pas avoir de représentations trop psychédéliques (section 1.2 page 2) nous commençons par translater les points pour que l'origine soit à la valeur minimale de x , y et z .

```
df <- data.frame(x = pcloc$X, y = pcloc$Y, z = pcloc$Z)
df <- as.data.frame(apply(df, 2, \((x) x - min(x)\)))
summary(df)

      x          y          z
Min. : 0.000  Min. : 0.000  Min. : 0.000
1st Qu.: 5.203  1st Qu.: 6.062  1st Qu.: 1.136
Median :11.027  Median :11.003  Median :12.415
Mean   :10.351  Mean   :10.597  Mean   :13.459
3rd Qu.:15.319  3rd Qu.:15.341  3rd Qu.:26.046
Max.   :19.800  Max.   :20.000  Max.   :33.652
```

POUR détouurer le houppier nous vous proposons la fonction artisanale `gomme()` définie ci-après. Cette fonction permet de censurer des points en sélectionnant une région rectangulaire dans la fenêtre de visualisation 3D. Son comportement se contrôle via un dialogue dans la console `R` qui est : `Next?` (`gomme/undo/fin`). Entrez `fin` quand le détourage est terminé, `undo` pour annuler le dernier gommage de points et `gomme` (ou rien du tout juste la touche « entrée ») pour sélectionner des points à supprimer. Notez que vous pouvez taper plusieurs fois la touche « entrée » pour faire plusieurs opération de gommage sans avoir à repasser par la console.

```
gomme <- function(df, censored = NULL, reponse = TRUE, size = 0.1, ncols = 255){
  if(is.null(censored)) censored <- logical(nrow(df)) # FALSE par défaut
  xlim <- range(df[, 1]) ; ylim <- range(df[, 2]) ; zlim <- range(df[, 3])
  zfac <- cut(df[, 3], breaks = ncols)
  df$col <- lidR::height.colors(ncols)[zfac]
  open3d()
  points3d(df[!censored, ], xlim = xlim, ylim = ylim,
            zlim = zlim, size = size, col = df$col[!censored])
  bbox3d()
  bck <- censored
  while(!is.na(reponse)){
    if(reponse){ # gomme
      bck <- censored
      f <- select3d()
      if(!is.null(f)){
        selected <- f(df[!censored, 1], df[!censored, 2], df[!censored, 3])
        censored[!censored][selected] <- TRUE
      }
    } else { # undo
      censored <- bck
    }
    pop3d()
    points3d(df[!censored, ], xlim = c(0, 100), ylim = c(0, 100),
              zlim = zlim, size = size, col = df$col[!censored])
    reponse <- askYesNo("Next?", prompts = c("gomme", "undo", "fin"))
  }
  return(censored)
}
cens <- gomme(df)
```

LA fonction `gomme()` renvoie un vecteur logique indiquant quels sont les points que vous avez gommés. Faites-vous plaisir après avoir fini le détourage en visualisant en 3D l'arbre n° 305 avec la code ci-après, vous devriez obtenir quelque chose de ressemblant à la figure 13 page 22.

```
plot3d(df, col = ifelse(cens, "grey", "red"), size = 0.1)
```

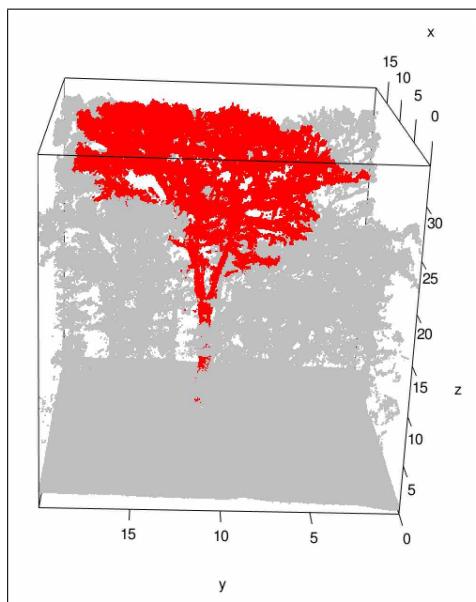
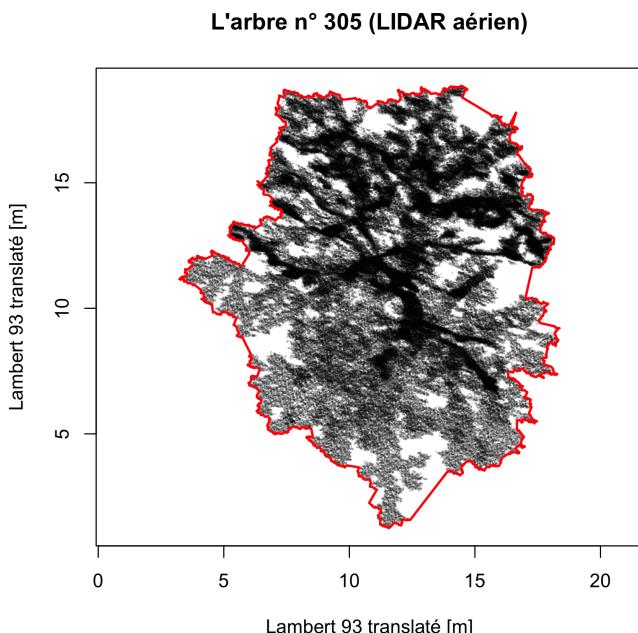


FIGURE 13 – Mise en évidence en rouge de l'arbre n° 305 après détourage manuel des données LIDAR aériennes.

POUR calculer la surface du houppier, commencez par visualiser le nuage de points de l'arbre n° 305 par en dessus avec des points semi-transparent. Ajoutez ensuite en rouge le polygone concave de détourage produit par le paquet *concaveman* [5].

```
dfa <- df[!cens, ]
points <- as.matrix(dfa[, 1:2])
plot(dfa$x, dfa$y, pch = ".", asp = 1, col = rgb(0, 0, 0, 0.2),
     xlab = "Lambert 93 translaté [m]", ylab = "Lambert 93 translaté [m]",
     main = "L'arbre n° 305 (LIDAR aérien)")
resc <- concaveman(points)
polygon(resc, col = "transparent", border = "red", lwd = 2)
```



La fonction `expanse()` du paquet `terra` [6] permet de calculer la surface d'un polygone. Utilisez-la pour calculer la surface du houppier que vous venez de détourer.

```
ressp <- terra::vect(resc, crs = "local", type = "polygon")
terra::expanse(ressp, transform = FALSE)
[1] 165.0001
```

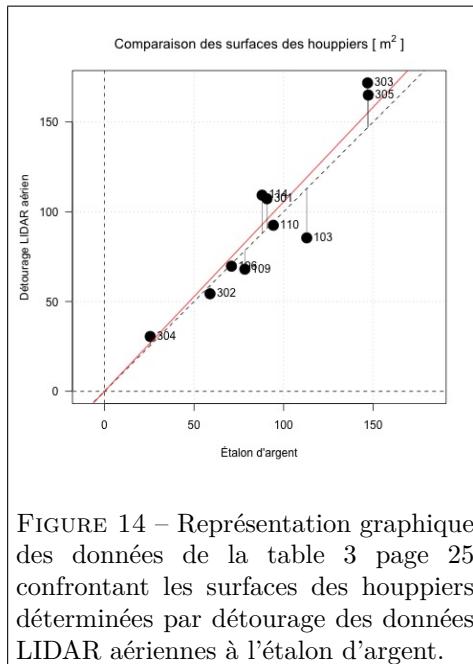
Vous devriez obtenir une valeur proche de celle pour l'arbre n° 305 dans la table 3 page 25, soit 165 m² environ. La figure 14 page 24 montre que les données LIDAR aériennes ont permis de corriger le point aberrant pour l'arbre n° 304 (comparer avec la figure 7 page 11). Il est vrai que sur l'orthophotographie d'automne il était bien difficile de distinguer son houppier de celui de son voisin tant ils ont une phénologie de la sénescence foliaire proches (figure 26 page 35).

3.2 Détourage avec données LIDAR terrestres

COMMENCEZ par récupérer le nuage de points au voisinage de l'arbre n° 305, il y a environ deux million de points sur une surface de 400 m² et les coordonnées GPS sont converties ici en LAMBERT 93 translatées comme pour les données LIDAR aériennes.

```
zoom <- terra::vect("HouppierLIDAR/zoom_CHS01_305/")
zoomL93 <- project(zoom, "EPSG:2154") # Lambert 93
df <- data.frame(x = geom(zoomL93)[ , "x"],
                  y = geom(zoomL93)[ , "y"],
                  z = zoom$z)
df <- as.data.frame(apply(df, 2, \((x) x - min(x)\)))
summary(df)

      x          y          z
Min. : 0.000  Min. : 0.000  Min. : 0.000
1st Qu.: 7.483  1st Qu.: 5.296  1st Qu.: 6.651
```



Median : 11.058	Median : 9.892	Median : 12.943
Mean : 11.101	Mean : 10.045	Mean : 14.115
3rd Qu.: 16.140	3rd Qu.: 14.590	3rd Qu.: 20.920
Max. : 20.533	Max. : 20.532	Max. : 33.554

UTILISEZ la fonction `gomme()` comme précédemment pour détourer le houppier de l'arbre.

```
cens <- gomme(df)
```

FAITES-vous plaisir après avoir fini le détourage en visualisant en 3D l'arbre n° 305 avec la code ci-après, vous devriez obtenir quelque chose de ressemblant à la figure 15 page 26.

TODO : discuter de la structure fractale ? Unicité de l'enveloppe convexe versus multiplicité des enveloppes concaves. Arbitraire du paramètre de concavité (faire faire manips?). Retour vers les orthophotos du printemps pour trouver un compromis raisonnable (les trous seront comblés par les feuilles).

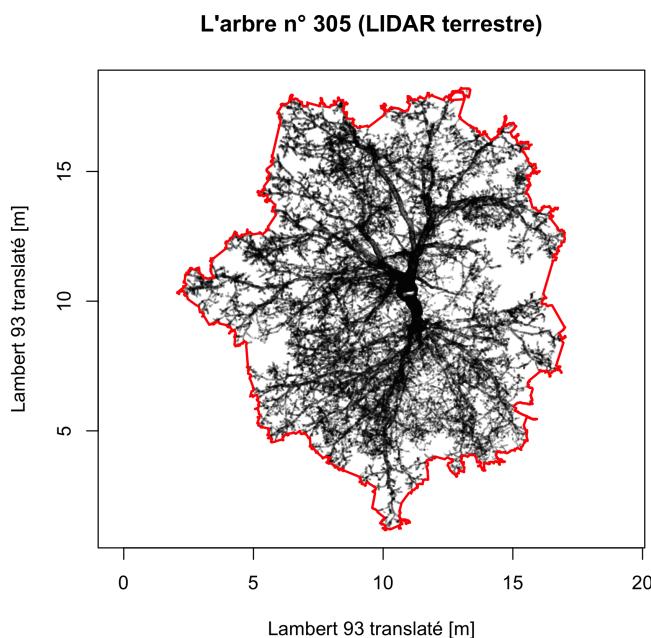
```
plot3d(df, col = ifelse(cens, "grey", "red"), size = 0.1)
```

DÉTOUREZ le houppier comme précédemment avec la fonction `concaveman()` du paquet éponyme.

```
dfa <- df[!cens, ]
points <- as.matrix(dfa[, 1:2])
plot(dfa$x, dfa$y, pch = ".", asp = 1, col = rgb(0, 0, 0, 0.1),
      xlab = "Lambert 93 traduit [m]", ylab = "Lambert 93 traduit [m]",
      main = "L'arbre n° 305 (LIDAR terrestre)")
resc <- concaveman(points)
polygon(resc, col = "transparent", border = "red", lwd = 2)
```

Arbre	Ref	Det	Écart abs.	Écart rel.
106	70.88	69.678	-1.204	-1.699
110	94.25	92.468	-1.780	-1.888
302	58.90	54.319	-4.586	-7.785
305	147.26	165.000	17.738	12.045
109	78.34	67.999	-10.345	-13.204
303	146.87	171.766	24.897	16.952
301	90.71	107.240	16.527	18.219
304	25.53	30.522	4.996	19.574
114	87.96	109.240	21.275	24.186
103	112.90	85.448	-27.453	-24.316

TABLE 3 – Les surfaces des houppiers en m^2 avec dans l'ordre la valeur de référence, celle déterminée par détournage des données LIDAR aériennes, l'écart entre les deux en absolu et relatif. Pour la visualisation de ces données voir la figure 14 page 24.



UTILISEZ la fonction `expande()` du paquet `terra` [6] pour calculer la surface du houppier que vous venez de détourner.

```
ressp <- terra::vect(resc, crs = "local", type = "polygon")
terra::expande(ressp, transform = FALSE)
[1] 154.2624
```

VOUS devriez obtenir une valeur proche de celle pour l'arbre n° 305 dans la table 4 page 27, soit $154 m^2$ environ. Si on compare les surfaces des houppiers issues des données terrestres et aériennes (figure 16 page 26) on constate qu'elles sont parfaitement cohérentes.

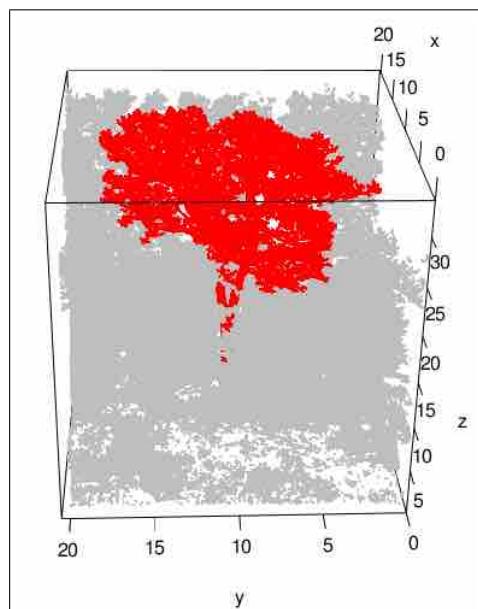


FIGURE 15 – Mise en évidence en rouge de l’arbre n° 305 après détourage manuel des données LIDAR terrestres.

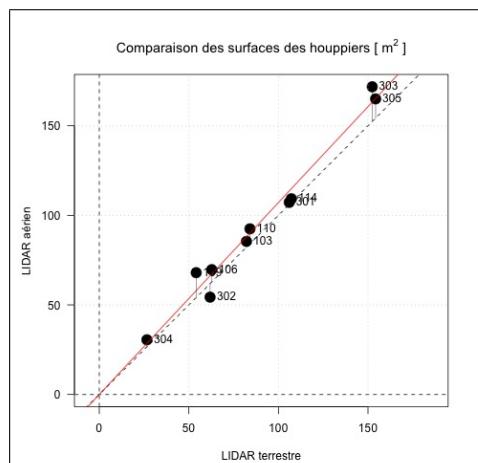


FIGURE 16 – Représentation graphique des données de la table 4 page 27 et de la table 3 page 25 confrontant les surfaces des houppiers déterminées par détourage des données LIDAR terrestres et aériennes.

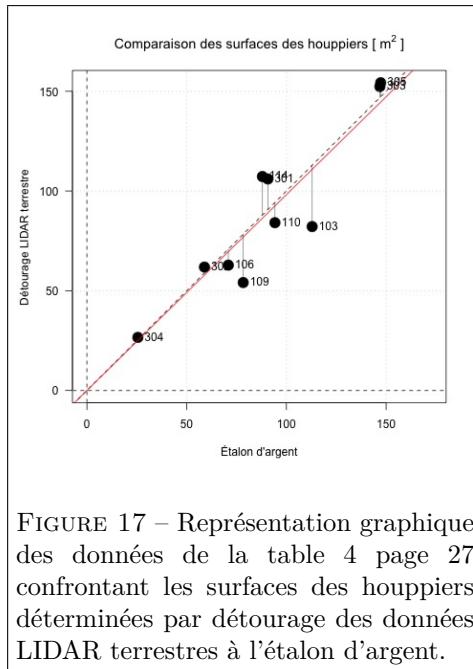


FIGURE 17 – Représentation graphique des données de la table 4 page 27 confrontant les surfaces des houppiers déterminées par détourage des données LIDAR terrestres à l'étalon d'argent.

Arbre	Ref	Det	Écart abs.	Écart rel.
303	146.87	152.355	5.485	3.735
304	25.53	26.595	1.070	4.192
305	147.26	154.262	7.000	4.754
302	58.90	61.860	2.955	5.016
110	94.25	84.113	-10.135	-10.753
106	70.88	62.828	-8.054	-11.363
301	90.71	106.044	15.330	16.900
114	87.96	107.256	19.291	21.931
103	112.90	82.167	-30.734	-27.222
109	78.34	54.107	-24.237	-30.936

TABLE 4 – Les surfaces des houppiers en m^2 avec dans l'ordre la valeur de référence, celle déterminée par détourage des données LIDAR terrestres, l'écart entre les deux en absolu et relatif. Pour la visualisation de ces données voir la figure 17 page 27.

4 Annexes

4.1 Remerciements

La réalisation de cette fiche de TD n'aurait pas été possible sans le concours de François DÉBIAS, Louis REY et Laurent MALABEUX, qu'ils en soient chaleureusement remerciés. Les orthophotographies sont de François DÉBIAS, assistant ingénieur de terrain au laboratoire de biométrie et biologie évolutive (université LYON 1, CNRS UMR 5558). Les données LIDAR aériennes sont de Louis REY, ingénieur d'étude de la plateforme ISIG (Ingénierie Spatiale, Images & Géomatique), CNRS UMR 5600, ENS de LYON. Les données LIDAR au sol sont de Laurent MALABEUX, technicien ONF du réseau recherche développement et innovation du pôle de CHAMBERY.

4.2 Importation des données LIDAR aériennes

Les données LIDAR peuvent être assez volumineuses, par exemple le fichier de départ⁶ pèse plus de 14 Gio sur disque, trop lourd pour être mis à disposition sur le serveur pédagogique. Mais pas de panique, nous allons extraire pour vous des sous-ensembles de ces données d'une taille suffisamment petite pour être manipulables par des ordinateurs de classe « bibliothèque universitaire ». Pour pouvoir le lire avec la fonction `readLAS()` du paquet `lidR` [9, 8] j'ai été obligé d'augmenter la valeur de ma variable `R_MAX_VSIZE` dans mon fichier `~/.Renviron` :

```
R_MAX_VSIZE=64Gb
```

J'ai utilisé la fonction avec le filtre `xyz` pour ne charger que le strict minimum en mémoire, c'est à dire les coordonnées (`x`, `y`, `z`) dans la zone qui m'intéresse. Cette opération dure un peu plus de trois minutes sur mon MacBook Pro et produit un objet `point_cloud` de type `LAS` de 3 Gio. Il peut y avoir des points aberrants dans le jeu de données, par exemple si le laser du drone-LIDAR a tapé un oiseau ou un insecte volant au dessus de la canopée générant un point anormalement haut. On en profite donc ici pour faire un peu de ménage avec les fonctions `classify_noise()` et `filter_poi()`. J'ai sauvegardé l'objet `point_cloud` au format `XDR` [11], cela m'a généré un fichier de 1 Gio sur disque qui demande 8 secondes pour être chargé en mémoire.

```
library(lidR) ; library(terra)
load("HouppierLIDAR/myextGPS.Rda")
mye <- vect(cbind(myextGPS[1:2], myextGPS[3:4]), crs = "EPSG:4326")
myeL93 <- project(mye, "EPSG:2154")
myfilter <- "-keep_xyz 872635.1 6565691 200 872744.6 6565892 400"
# (min_x min_y min_z max_x max_y max_z)
point_cloud <- readLAS("cloud_47c18ab99f219994.las", select = "xyz", filter = myfilter)
point_cloud <- classify_noise(point_cloud, ivf(res = 1))
point_cloud <- filter_poi(point_cloud, Classification != LASNOISE)
save(point_cloud, file = "point_cloud.Rda")
```

Le code suivant a été utilisé pour produire la carte de surface de la zone d'intérêt.

⁶. `cloud_47c18ab99f219994.las` pour les données d'une partie de la parcelle C3 de la forêt de Seillon survolée par drone-LIDAR le 2024-03-04

```
dsm <- rasterize_canopy(point_cloud, res = 0.05, algorithm = p2r())
dsmGPS <- terra::project(dsm, "EPSG:4326")
writeRaster(dsmGPS, file = "HouppierLIDAR/dsmGPS.tif")
```

Le code suivant a été utilisé pour produire la carte de l'altitude du sol.

```
sol <- classify_ground(point_cloud,
  algorithm = csf(rigidity = 2, cloth_resolution = 1), last_returns = FALSE)
sol <- sol[sol$Classification == 2, ] # points du sol
dtm <- rasterize_terrain(sol, res = 0.05, algorithm = tin())
dtmGPS <- terra::project(dtm, "EPSG:4326")
writeRaster(dtmGPS, file = "HouppierLIDAR/dtmGPS.tif")
```

Le code suivant a été utilisé pour extraire les nuages de points au voisinage des arbres d'intérêt.

```
## load("point_cloud.Rda")
pclocTree <- function(the_tree, dm = 10){
  centers <- vect("HouppierLIDAR/centers/")
  centersL93 <- project(centers, "EPSG:2154")
  xL93 <- geom(centersL93[names == the_tree,])[, "x"]
  yL93 <- geom(centersL93[names == the_tree,])[, "y"]
  pcloc <- clip_rectangle(point_cloud, xL93 - dm, yL93 - dm, xL93 + dm, yL93 + dm)
  save(pcloc, file = paste0("HouppierLIDAR/pcloc_", the_tree, ".Rda"))
}
for(the_tree in allTrees) pclocTree(the_tree)
```

4.3 Importation des données LIDAR terrestres

LDONNÉES originelles se présentent sous la forme de deux fichiers⁷ de l'ordre de 1 Gio chacun, un pour les arbres de la zone nord et un pour les arbres de la zone sud, déjà pré-traités pour en exclure les points du sol. Les coordonnées des points utilisent un référencement relatif, et les coordonnées des tiges des arbres d'intérêt ont été relevées avec un pointeur laser :

```
library(readODS)
locarbres <- read_ods("HouppierLIDAR/locarbres.ods", as_tibble = FALSE)
locarbres <- subset(locarbres, Arbre != 110)
locarbres[ , 1:5]

Arbre zone   x     y     z
1    109    N 13.380 37.467 -0.603
3    114    N -16.188 -9.609  0.034
4    301    N -32.600 25.133  0.264
5    302    N -15.207 39.562 -0.131
6    303    N -26.600 -26.739  0.181
7    103    S 15.204 74.731 -0.240
8    106    S -14.716 26.227 -1.125
9    304    S   4.109 11.086 -0.288
10   305    S -38.126 38.777 -1.914
```

J'AI exclu l'arbre n° 110 parce que je n'avais pas sa localisation précise mais juste un positionnement approximatif à 10 m au nord du n° 109 (voir la figure 4 page 7). Comme on connaît la localisation GPS précise des arbres d'intérêt (table 1 page 5), la question qui se pose est de savoir comment convertir les coordonnées du référencement relatif en coordonnées GPS. Je me suis grandement inspiré d'un post sur *StackExchange*⁸. Il faut évidemment traiter la zone nord et la zone sud séparément, je ne donne ici que le code pour la zone nord. On commence par importer les données dans **R**, toujours à l'aide du paquet **lidR** [9, 8] :

7. Les fichiers CHS01_EXT_Nord_Vegetation_2cm.laz et CHS01_Ext_Sud_VEGETATION_2cm.laz
8. <https://gis.stackexchange.com/questions/23998/seeking-qgis-affine-equation>

```
library(lidR)
pathN <- "scan_terreste/zone1/LAZ_Zone_Nord/CHS01_EXT_Nord_Vegetation_2cm.laz"
lazN <- readLAS(pathN, select = "xyz")
```

CETTE opération prend environ 40 secondes sur mon ordinateur et génère un objet de type LAS de 1.5 Gio. Je fais ensuite la jointure avec la table des coordonnées GPS de référence des arbres de la zone nord.

```
GPS <- read_ods("HouppierLIDAR/GPS.ods", as_tibble = FALSE)
GPS$ID <- substr(GPS$ID, 7, 9) # pour enlever le "CHS01_"
cmp <- merge(GPS[ , -2], locarbres[ , 1:5], by.x = "ID", by.y = "Arbre")
cmp <- subset(cmp, zone == "N")[ , c(1:3, 6:7)] # colonnes utiles zone nord
cmp
   ID      x.x     y.x     x.y     y.y
3 109 5.238944 46.17034 13.380 37.467
4 114 5.238232 46.17042 -16.188 -9.609
5 301 5.238558 46.17068 -32.600 25.133
6 302 5.238822 46.17059 -15.207 39.562
7 303 5.237997 46.17043 -26.600 -26.739
```

ON veut donc convertir les coordonnées du référencement local **x.y** et **y.y** en coordonnées GPS **x.x** et **y.x**. Je range dans les matrices **source** et **dest** les coordonnées des arbres dans les deux systèmes.

```
source <- with(cmp, cbind(x.y, y.y)) ; colnames(source) <- c("x", "y")
source
   x     y
[1,] 13.380 37.467
[2,] -16.188 -9.609
[3,] -32.600 25.133
[4,] -15.207 39.562
[5,] -26.600 -26.739

dest <- with(cmp, cbind(x.x, y.x)) ; colnames(dest) <- c("x", "y")
dest
   x     y
[1,] 5.238944 46.17034
[2,] 5.238232 46.17042
[3,] 5.238558 46.17068
[4,] 5.238822 46.17059
[5,] 5.237997 46.17043
```

ON utilise alors un bête modèle linéaire pour passer de la source à la destination. Dans **R** c'est d'une compacité étonnante, mais bon, c'est un peu son cœur de métier aussi.

```
lmN <- lm(dest ~ source)
lmN$coefficients
            x           y
(Intercept) 5.238438e+00 4.617031e+01
sourcex      5.199400e-06 -8.345193e-06
sourcey      1.161513e-05  3.647159e-06
save(lmN, file = "HouppierLIDAR/lmN.Rda")
```

LES nouvelles coordonnées **x** et **y** se calculent donc comme la somme d'une constante (**Intercept**) et d'une combinaison linéaire des anciennes coordonnées **sourcex** et **sourcey**. La multiplication matricielle de **R** est alors d'une redoutable efficacité, une poignée de secondes pour calculer les coordonnées de près de 62 millions de points. Il faut juste bien penser à ajouter une colonne de 1 pour intégrer la constante.

```
new <- cbind(1, lazN$x, lazN$y) %*% lmN$coefficients
```

RESTE à sauvegarder le résultat. J'ai d'abord pensé à utiliser un objet de type **LAS** mais j'ai eu un *warning* comme quoi ce n'était pas conseillé pour des coordonnées GPS. Je me suis donc tourné vers un objet de type **spatVect** en portant l'altitude dans la couche **z**. Là les choses se gâtent, il faut compter 40 secondes pour créer l'objet **lazNgps**, près de 3 minutes pour peupler sa couche **z**, 18 minutes pour se restreindre à la région utile et 8 minutes pour sauvegarder le tout sur disque.

```
lazNgps <- vect(new, crs = "EPSG:4326")
lazNgps$z <- lazN$Z
load("HouppierLIDAR/myextGPS.Rda")
lazNgps <- crop(lazNgps, ext(myextGPS))
writeVector(lazNgps, "lazNgps")
```

LE fichier pèse ses 3 Gio sur disque, prend 4 minutes à être chargé et me mange 15 Gio de RAM, c'est beaucoup trop lourd pour une exploitation souple, on va donc extraire les régions autour des arbres d'intérêt. J'ai utilisé ici une région carrée de 20 m de coté, soit 400 m² de surface, centrée sur la position GPS des arbres.

```
library(readODS) ; library(terra)
#lazNgps <- vect("lazNgps")
GPS <- read_ods("HouppierLIDAR/GPS.ods", as_tibble = FALSE)
locarbres <- read_ods("HouppierLIDAR/locarbres.ods", as_tibble = FALSE)
cropTree <- function(the_tree, buf_m = 10){
  mye <- with(subset(GPS, ID == the_tree),
              ext(buffer(vect(cbind(x, y), crs = "EPSG:4326"), buf_m)))
  zoom <- crop(lazNgps, mye)
  writeVector(zoom, paste0("HouppierLIDAR/zoom_", the_tree))
}
nordTrees <- paste0("CHS01_", subset(locarbres, zone == "N")$Arbre)
for(the_tree in nordTrees){
  print(the_tree)
  cropTree(the_tree)
}
```

ON a maintenant des fichiers d'une taille plus raisonnable, de l'ordre de 150 Mio par arbre sur disque, chargés en 10 secondes en mémoire. Ce sont là des fichiers que l'on peut mettre à la disposition des étudiants sur le serveur pédagogique.



FIGURE 18 – détourage de l'arbre n° 103



FIGURE 19 – détourage de l'arbre n° 106

4.4 Détourage des houppiers sur orthophotographies automnales



FIGURE 20 – détournage de l'arbre n° 109



FIGURE 21 – détournage de l'arbre n° 110



FIGURE 22 – détournage de l'arbre n° 114



FIGURE 23 – détourage de l'arbre n° 301



FIGURE 24 – détourage de l'arbre n° 302



FIGURE 25 – détourage de l'arbre n° 303



FIGURE 26 – détourage de l'arbre n° 304



FIGURE 27 – détourage de l'arbre n° 305

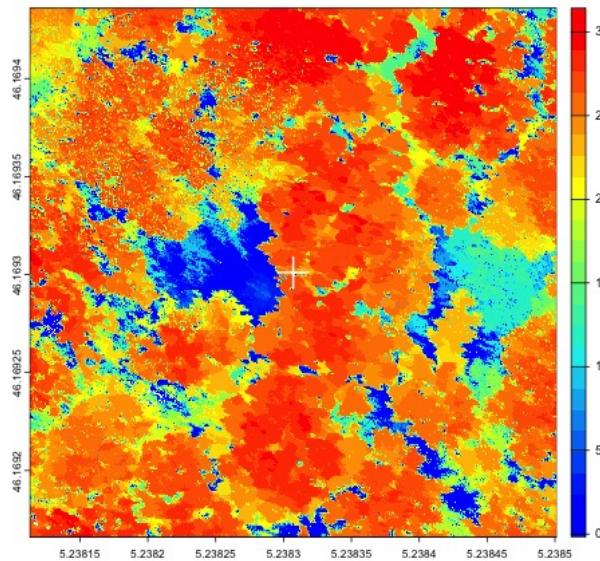


FIGURE 28 – Canopée de l'arbre n° 103

4.5 Zoom sur la canopée des houppiers d'intérêt

```
zoomCannop <- function(the_tree, pm = 15){
  require(terra) ; require(lidR)
  dsmGPS <- rast("HouppierLIDAR/dsmGPS.tif")
  dtmGPS <- rast("HouppierLIDAR/dtmGPS.tif")
  chmGPS <- dsmGPS - dtmGPS
  centers <- vect("HouppierLIDAR/centers")
  x <- geom(centers[centers$names == the_tree, ])[ , "x"]
  y <- geom(centers[centers$names == the_tree, ])[ , "y"]
  mypt <- vect(cbind(x, y), crs = "EPSG:4326")
  myext <- ext(buffer(mypt, pm))
  plot(chmGPS, col = height.colors(25), ext = myext)
  points(mypt, cex = 3, pch = 3, col = "white", lwd = 2)
}
```

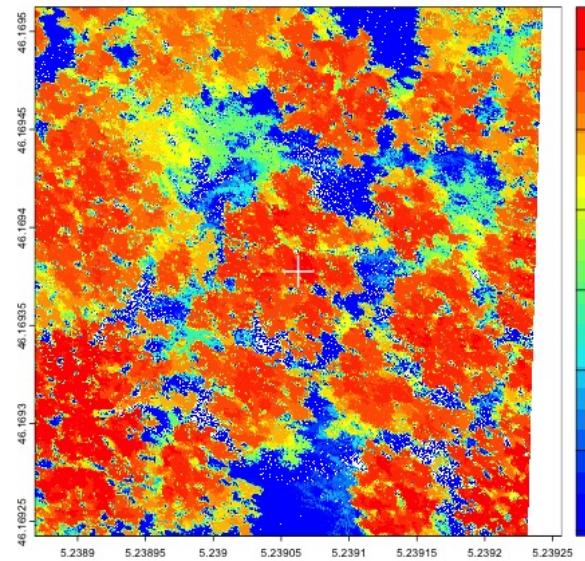


FIGURE 29 – Canopée de l'arbre n° 106

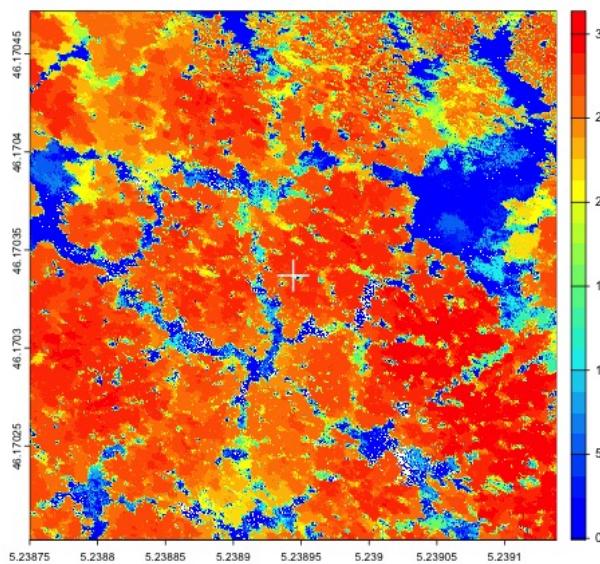


FIGURE 30 – Canopée de l'arbre n° 109

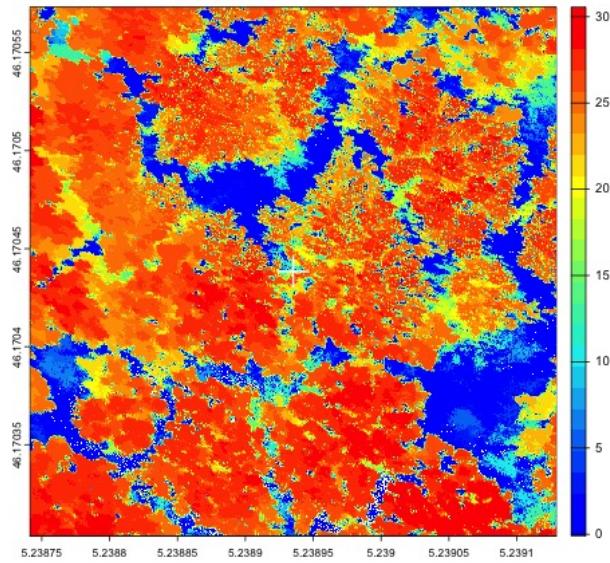


FIGURE 31 – Canopée de l'arbre n° 110

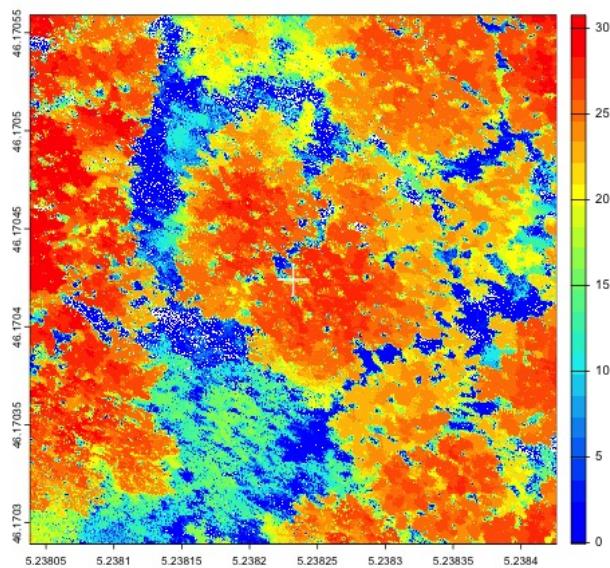


FIGURE 32 – Canopée de l'arbre n° 114

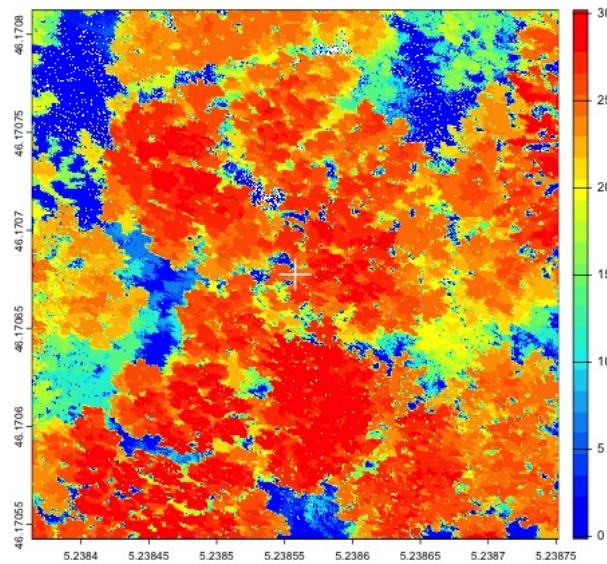


FIGURE 33 – Canopée de l'arbre n° 301

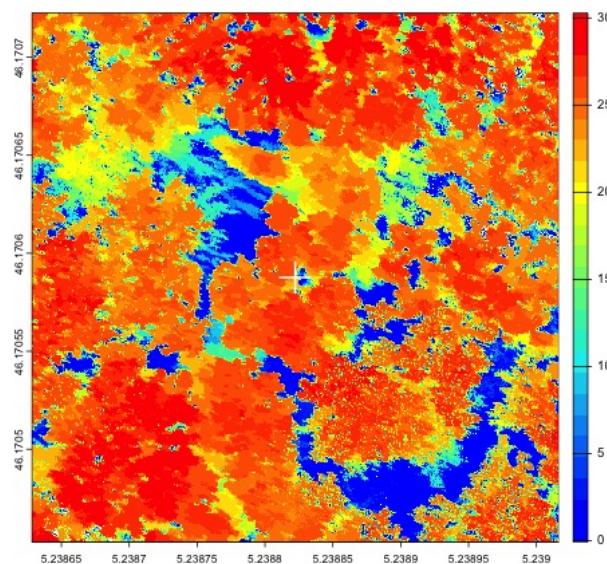


FIGURE 34 – Canopée de l'arbre n° 302

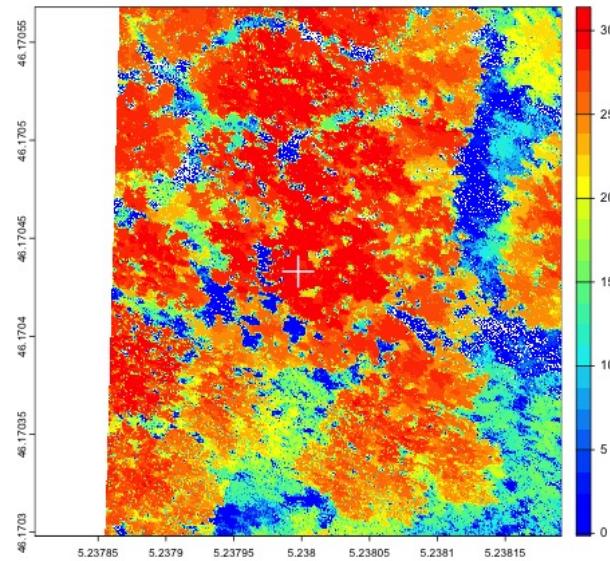


FIGURE 35 – Canopée de l'arbre n° 303

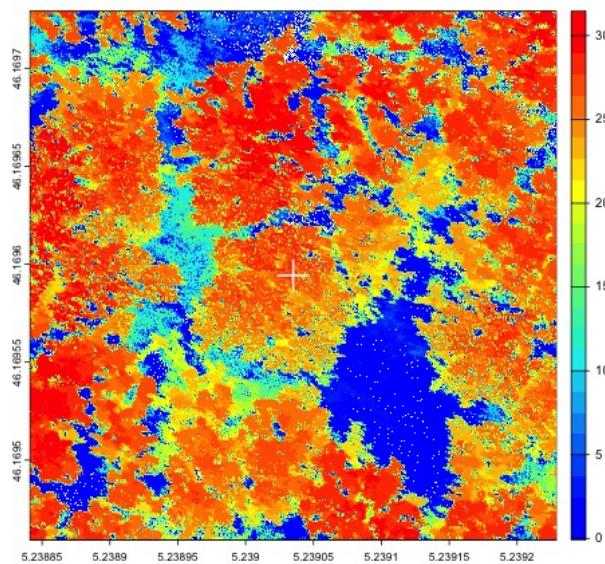


FIGURE 36 – Canopée de l'arbre n° 304

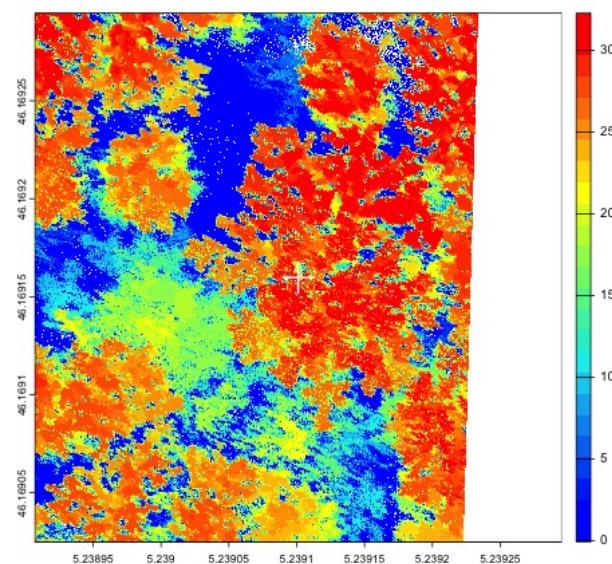


FIGURE 37 – Canopée de l'arbre n° 305

Références

- [1] D. Adler and D. Murdoch. *rgl : 3D visualization device system (OpenGL)*, 2014. R package version 0.93.996.
- [2] J. André. *Petites leçons de typographie*. Editions du jobet, IRISA, 2003-2020.
- [3] C. Barthod. Le système de surveillance de l'état sanitaire de la forêt en France. *Revue forestière française*, 46 :564–571, 1994.
- [4] S. Dumas and D. Pivot. Forêt domaniale de SEILLON. surface : 613,68 ha. révision d'aménagement forestier 2009-2028. Technical report, Office national des Forêts, Direction territoriale Rhône-Alpes, Agence Ain Loire Rhône, 2008. 65 pp.
- [5] J. Gombin, R. Vaidyanathan, and V. Agafonkin. *concaveman : A Very Fast 2D Concave Hull Algorithm*, 2020. R package version 1.1.0.
- [6] R.J. Hijmans. *terra : Spatial Data Analysis*, 2024. R package version 1.7-71.
- [7] F. Lebourgeois, J. Differt, A. Granier, N. Bréda, and E. Ulrich. Premières observations phénologiques des peuplements du réseau national de suivi à long terme des écosystèmes forestiers (RENECOFOR). *Revue forestière française*, 54(5) :407–418, 2002.
- [8] J.-R. Roussel and D. Auty. *Airborne LiDAR Data Manipulation and Visualization for Forestry Applications*, 2025. R package version 4.1.2.
- [9] J.-R. Roussel, D. Auty, N.C. Coops, P. Tompalski, T.R.H. Goodbody, A. Sánchez Meador, J.-F. Bourdon, F. de Boissieu, and A. Achim. *lidR : An R package for analysis of Airborne Laser Scanning (ALS) data*. *Remote Sensing of Environment*, 251 :112061, 2020.
- [10] J.-R. Roussel and J. Qi. *RCSF : Airborne LiDAR Filtering Method Based on Cloth Simulation*, 2020. R package version 1.0.2.
- [11] Sun Microsystems. XDR : external data representation standard. RFC 1014. Technical report, Network Working Group, 1987.
- [12] E. Ulrich. Le réseau RENECOFOR : objectifs et réalisation. *Revue forestière française*, 47 :107–124, 1995.