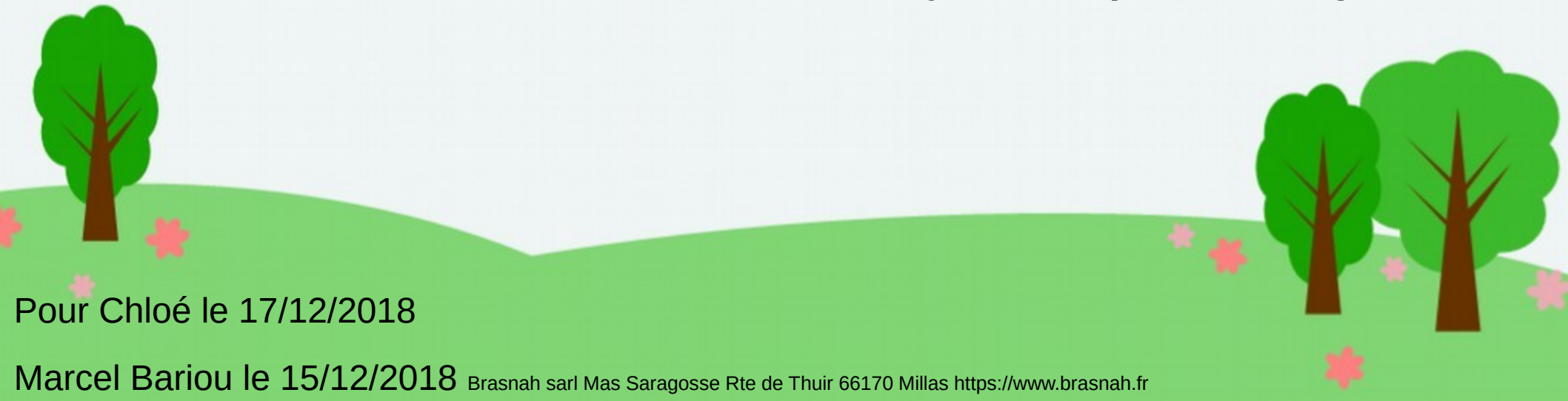


Tomographie sur Image Lidar Forestière (TILF)

**Usage de PDAL, PCL, QGIS, LIBLAS
C++, C_CUDA, PYCUDA, PYQGIS
en traitement de mesures forestières
à base de relevés LIDAR**

Toutes les librairies employées relèvent de diverses licences de logiciels libres de droit

Mesurer, analyser, comprendre & agir



Pour Chloé le 17/12/2018

Marcel Bariou le 15/12/2018 Brasnah sarl Mas Saragosse Rte de Thuir 66170 Millas <https://www.brasnah.fr>

Le programme de travail



- Les origines de la démarche
- Les fichiers de travail
- Environnement technologique et installations
- Ordonnancement industrialisation Python, C++
- 1^{er} coeur de bibliothèques PDAL, PCL, CC
- 2ème coeur de bibliothèques QGIS, Sp*lite, PG
- 3ème coeur de bibliothèque Sklearn, PyTorch



Les objectifs



- Mettre en œuvre les outils d'acquisition
- Qu'extrait-on d'un nuage de points ?
- Mise en œuvre des outils de traitement
- Apprendre & comprendre
- Exploiter les résultats

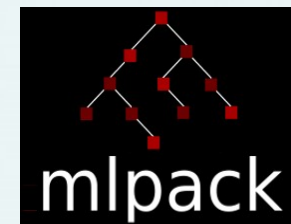


Crédits de cette présentation



- Inspiré des travaux et ateliers de :
- Howard Butler (Hobu Inc, Tweets @ <http://twitter.com/howardbutler>)
- Michaël Smith (USACE CRREL, Tweets @ http://twitter.com/michael_smith)

15 ans de R&D logiciel SIG et LIDAR



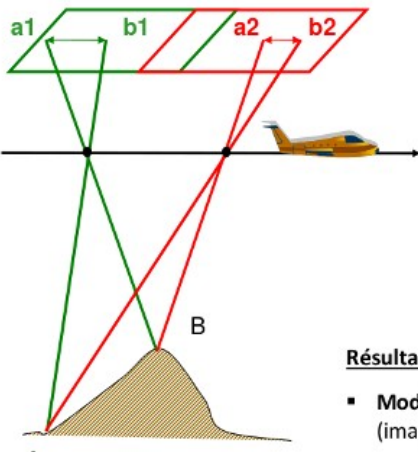
Le LIDAR



- Permet des mesures de distance par laser
- Monté sur véhicule porteur (T.A.M) ou en mains
- Géolocalise des points dans un repère 3D

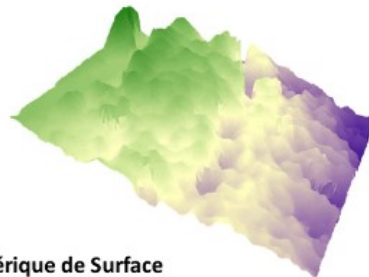
AVANT : photogrammétrie

Points homologues entre les 2 images:
→ a1 et a2 ; b1 et b2



✓ Un même objet du territoire est vu depuis plusieurs points, sous des angles différents

- Recherche automatique des points homologues (corrélation)
- Calcul des altitudes correspondantes



Résultat =

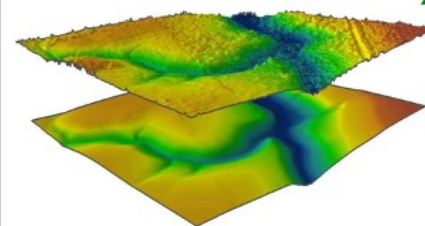
- **Modèle Numérique de Surface**
(image représentant les altitudes de la canopée)

Maintenant : Lidar

Exemple : mission LIDAR /Combe d'Aillons
Surface couverte ~ 2 500 ha

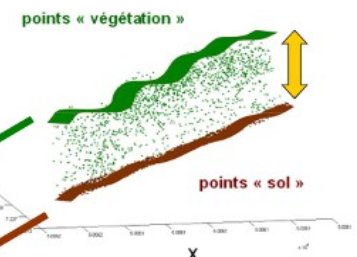
Nuage de points bruts 3D =
417 millions de points
16 points / m2 en moyenne (sol+végétation)
(après fusion des bandes de vol)

MNS: modèle numérique de surface



MNT : Modèle numérique de terrain

Classification des points :



MNH: Modèle Numérique de Hauteur

Fonctionne comment ?



- Scanner à base de laser (250 nm à 2 μ)
- Horloge à très haute précision
- Système de positionnement par satellite (G.P.S)
- Centrale Inertielle (T, R, L), accéléro. (3 axes)
- Stockage de données
- Mesure de temps de vol + Réf. Précise au sol
- Si plusieurs échos classification grossière





Type de lidar

- Aérien (A.L.S),
- Terrestre (T.L.S) mobile (Véh., pers.) ou fixe,
- Bathymétrie.

LIDAR GEOSLAM utilisé par EXAMETRICS

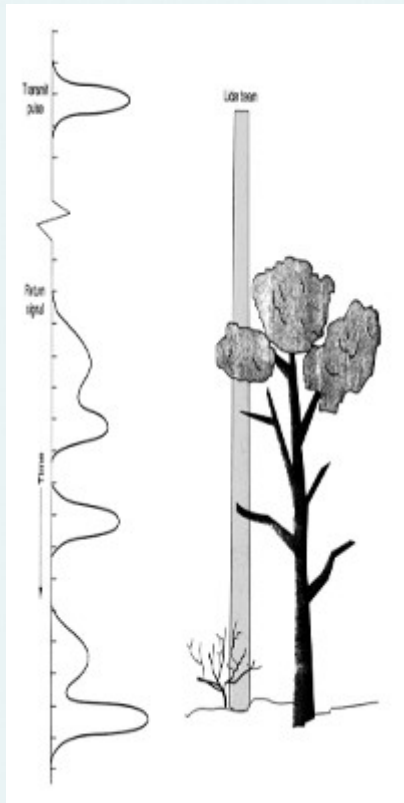


- $\lambda = 905 \text{ nm}$, poids 1 Kg / déplacement : 3 à 4 km/h
- Zone de balayage éventail plan ouvert à 270° .
- Rotation éventail (2 Hz) volumique ($270^\circ \times 360^\circ$)
- Résolution 0.625° , soit 432 angles de tir
- Le balayage de l'éventail se fait en 10ms (100 Hz),
- Une cadence d'acquisition de 43 200 points par seconde
- Ambiguïté distance 3500 m (effacé par atténuation)
- 1 seul écho pas de classification





Analyse du train d'onde



- Image ci-contre => échos multiples
- Sintegra (multiple) Exametrics (1 écho)
- Impulsionnel, continu, interféro.,
- Mesure de temps de vol CT/2,
- Orientation faisceau (T,R,L),
- Positionnement I.M.U, G.P.S
- Temps pour la trajectographie,
- Si centrale inertielle position réf. Inst.
- Post traitement => fichiers de points,
- Plusieurs fichiers de points (« registration / agrégation ») :
 - => recalage
 - => agrégation





La chaîne d'outils

- LIDAR est l'instrument de mesures,
- Données **abondantes** selon des standards ouverts,
- Logiciels de traitement (goulet d'étranglement) :
 - Propriétaires (non présentés, voir vendeurs car fermés),
 - Logiciel LIBRES (outils et bibliothèques)
 - Outils & Cadriciels : Computree, CloudCompare, ParaView, Mayavi,
 - Bibliothèques : PCL, VTK, PDAL (LibLas), LasTools, LasPy,
 - Besoins d'algorithmique géométrique et de puissance de calcul, parallélisation (« Threading »), processeurs graphiques (GPU, ex. : CUDA).



PDAL intro.



- **P**oint **D**ata **A**bstract **L**ibrary, à comparer à **GDAL** dans sa structure de mise en œuvre,
- Offre une voie de standardisation et répond à un besoin d'opérations de base,
- Les étapes :
 - Changement de format « translate »,
 - Les modes de traitements,
 - Usages.

<https://pdal.io> <http://github.com/PDAL/PDAL> Licence BSD Basé sur LIBLAS (2010)
<http://lidar.io> Projet USACE Grid Université du Kansas (USA)



Fondations



- Librairie C++, origine dans LibLas
- API Python étroitement liée à NUMPY,
- Les étapes de traitement rassemblent les fonctions de lecture, écriture, filtrage séquentiellement dans un pipeline,
- Le lecteur lit des données normalisées (~30 formats),
- Le scripteur absorbe les données au format (~15) voulu,
- Le filtre modifie les nuages, impression en fichier ~15 formats.



Environnement de travail



- **Objectifs** (traitement de nuages de points LIDAR) :
 - Possibilités de prototypage, souplesse de codage,
 - Limitation des conflits de bibliothèques,
 - Capacité de traitements rapides (locaux ou distants)
- **Outils de traitement de nuage de points lidar:**
 - Manipulation visuelle de nuage de points ex. : CloudCompare
 - Ligne de commande et bibliothèques : PCL, PDAL, LasTools, + spécifique
 - Linux/Unix (focus Ubuntu 18.04 LTS), Spyder 3.6, Python 3.6, Jupyter
 - Spécifique C++ / Python (Maîtrise, adapté, plus de V.A mais long devel.)
- **Outils de classification :**
 - Sklearn, PyTorch, MIPack, OpenCV



Technologies complémentaires



- Bases de données spatiales (options) :
 - PostGres/PostGis,
 - SpatiaLite (SQLite), RasterLite
 - MySQL
- Systèmes d'information géographique:
 - QGis
 - GRASS,
 - OGR
- Parallélisation de calcul



Script de base PyThon



- Python ordonnance les commandes en ligne

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Tue Dec 11 23:02:35 2018
@author: mbariou
"""
import os
import re
import subprocess
#os.system("pdal info --stats -i 12616_Massane_SemisBrut_L93-IGN69_000007.las")
from os import listdir
from os.path import isfile, join
onlyfiles = [f for f in listdir("./") if isfile(join(mypath, f))]
for f in onlyfiles :
    if re.search(".las", f) :
        print(f)
        #os.system("pdal info --stats -i "+ f)
        #os.system("pdal info " +f + " -p 0")
        #os.system("pdal info " +f + " --metadata")
        os.system("pdal info " +f + " --metadata |grep maxx")
        os.system("pdal info " +f + " --metadata |grep maxy")
        os.system('pdal info ' +f + ' --metadata More ? |jq ".metadata.maxx, .metadata.maxy, .metadata.minx, .metadata.miny"')
        varOutput = subprocess.check_output('pdal info ' +f + ' --metadata More ? |jq ".metadata.maxx, .metadata.maxy, .metadata.minx, .metadata.miny"', shell=True)
        print (varOutput)
        varOutput = os.system("pdal info " +f + " -p 0")
        print (varOutput)
```



Les étapes => le pipeline du processus



1 - Acquisition sur terrain, données LIDAR et amers géolocalisés (lots de mesures de 15 à 20 mn). [Lidar Geoslam +GPS]	7 - Nettoyage des fichiers PNG et fermeture (75%) des enveloppes convexes de sections d'arbre : opération <u>manuelle</u> , faiblesse algo, à approfondir. [GIMP]
2 - Construction du fichier *.las et validation, taille : 1 à 3 Go conditionnée par la durée de la mesure. [Logiciel Geoslam + CloudCompare]	8 - Erosion morphologique, normalisant la densité, sur l'image *.png. [GIMP, G'MIC]
3 - Extraction du modèle numérique de terrain (pas de 50 cm) sur lot. Après transformation éventuelle (11) elle constituera l'empreinte au sol en S.I.G. [CreateMNT_KNN_Optimal_CM]	9 - Application des batteries de filtres décroissants en taille (120 cm à 20 cm), détection des pivots d'accumulation, effacement des sections détectées. [FILT_CIRCLE_STANDARD]
4 - Mise des pieds d'arbre des lots au relief Zéro (orthoprojection par retrait MNT). [Substract_MNT_CM]	10 - Élimination des doublons (pivots accumulation), avec arbitrage de diamètre sélectionné, extraction Z arbre dans MNT. [FILT_CIRCLE_STANDARD]
5 - Extraction d'une section, par lot, de 10 cm d'épaisseur, à 1.30 m de hauteur, parallépipédique, parallèle au sol. [Prj_Extract_Tile]	11 - Calcul des matrices de transformation, à partir des amers sélectionnés sur chaque lot. [WG84_2_RFG93]
6 - A partir des tranches extraites des lots création de fichiers images (*.png) par orthoprojection sur XOY. Résolution de construction 1cm / 1 px. [Create_PNG-X-Y-Zmoy]	12 - Transformation C.R.S (WGS84 => RFG93) des sections détectées par les matrices précédentes pour chaque zone et préparation des données attributaires pour mise en S.I.G . [Forest_Data_For_SIG]

[*] => désigne outils existants ou utilitaires C++ développés spécifiquement





Ressources de mesures

- Fichier LIDAR¹ (E¹) terrestre global non géoréférencé (nécessite un découpage, calé sur le tuilage MNT [500m x 500m] et le tuilage moyenné indexé) peu précis
- Fichier Lidar MNT(S²) après tuilage moyenné de 5m x 5m ou 2m x 2m de mettre les pieds d'arbres à Z=0 mais interpolation peu précise
- Orthophoto TIFF² pour tentative extraction Houppier avec apprentissage + géométrie amers (**si identifiables dans nuages**)

Fournitures : ¹ => EXAMETRICS ² => Sintegra



Données LIDAR Exametrics



- Relevées en sous-bois potentiel important de données
- Un nuage global reconstitué, avec accumulation d'erreurs, référencé dans le seul instrument de mesure à relocaliser en RFG93 transformation Helmert (paramètres à calculer voir plus loin)
- Un nuage fractionnaire à identifier dans le précédent, a priori moins d'erreurs, recherche visuelle et corrélation
- Souffrent d'une absence de procédure de relevé comme de points de référence géolocalisés
- LIDAR à écho unique => pas de classification distance
- Abondance de points au sol (MNT possible?)



Données LIDAR Sintegra



- Relevé aérien, semis global décomposé en **MNC**, **MNS**, **MNT** (**C**anopée, **S**urface, **T**errain)
- Correctement géoréférencé, projeté
- Echos multiples donc classification **MNC**, **MNS** et **MNT** les fichiers *.xyz sont reconvertis en *.las
- Extraction par dalle de mesure 500m x 500m
- Interpolation imprécise en **MNT**



Orthophoto (*.tif) Sintegra



- Relevé aérien
- Résolution à 10 cm
- Seul point homologue identifiable (photo² (S), lidar^{1, 2}) => le refuge des Colomates.
- Tentative extraction des houppiers (OpenCV traitement image et classification) => non traité

Fournitures : ¹ => EXAMETRICS, ² => Sintegra



Apprendre à raisonner sur des mesures fausses



- Calcul empreinte du nuage (E^1) et Barycentre
- Droite de régression du nuage (E^1) sur empreinte au sol
- Barycentre relevé La Massane
- Droite de régression du relevé la Massane
- Identification position/orientation refuge Colomates (Images TIFF (S^2), à confirmer par mesure sur place
- Calcul paramètre de Helmert => ajustement grossier de (E^1) en RFG93 sur le site de La Massane
- Raffinement du positionnement du nuage (E^1)

Fournitures : ¹ => EXAMETRICS, ² => Sintegra



POINTS HOMOLOGUES



- En ortho-imagerie²,
- Nuage global¹
- Nuage cloture¹
- Nuage MNS²

Après extraction de tuiles 50m x 50m , l'identification de la géométrie de la bâtisse du refuge des Colomates, permet l'évaluation de la matrice de Helmert pour un recalage absolu grossier du nuage EXAMETRICS dans le référentiel RFG93.

Fournitures : ¹ => EXAMETRICS, ² => Sintegra





Stratégie de calcul

- Dans le même référentiel géodésique sur les objets homologues de deux nuages (1, 2):
 - extraire des segments au moins 2, non colinéaires, orientés à l'identique dans la zone de référence avec leurs homologues du nuage à recaler,
 - construire les plans dérivés (P1, P2) issus des vecteurs initiaux et homologues et orientés à l'identique deux à deux ,
 $\vec{N}_1$
 $\vec{N}_2$
 - Calculer les barycentres des triangles générateurs de plan,
 - Translater [T] vers le barycentre cible,
 - Calculer les normales $\vec{N}_1$ et $\vec{N}_2$
 - Paralléliser les normales orientées à l'identique, Matrice [A]
 - Évaluer la matrice de rotation autour de la normale Matrice [B]
 - En déduire la matrice d'Helmert (A, B, T)
 - Démarche classique **mais à bien maîtriser**



Outils de préparation



- FindPointXY.py (Recherche points bornés dans fichier xyz).
- BuildLasBoundaryAsJson.py (xyz → las) et extraction empreinte json avec pipeline PDAL et commande en ligne.
- WriteFileLinePerLine.py analyse/écriture sur fichier *.xyz et insertion d'en-tête.
- ScanForXYZ.py recherche d'en-tête ou de coordonnées par ligne dans un fichier XYZ
- PrepsTranslateRotate.py rotation translation
- CheckPyQtSIP.py validation compatibilité de version
- Transf3DLaMassanne.cu (C_Cuda Translation, rotation => 8.10^6 tâches //)
- ExtractTileFromlasFile.cpp (C++)



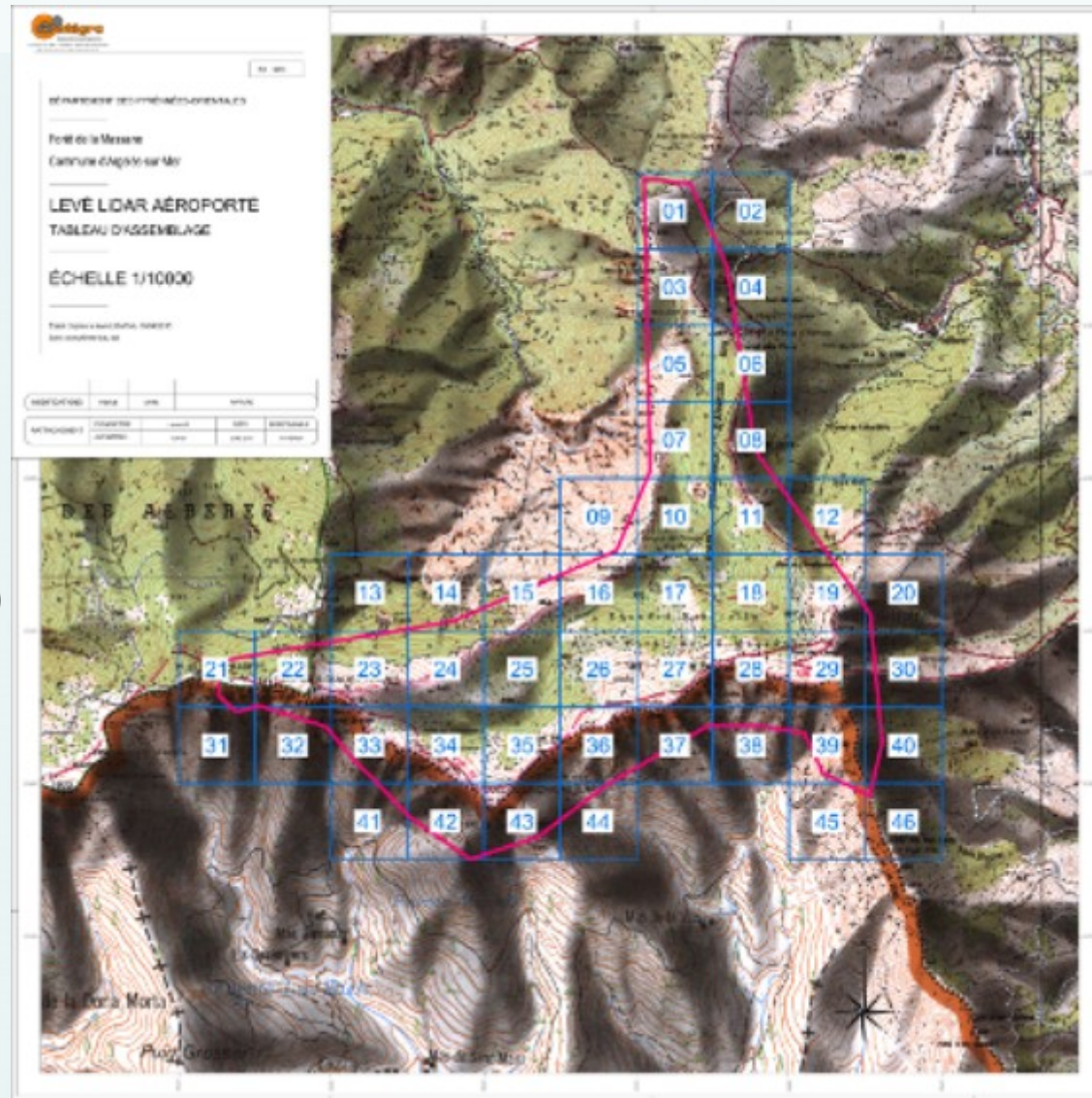
Zonage Photo & LIDAR



Crédits : Sintegra

Zone Initiale d'intérêt :

- 07
- Présence amer (refuge)
- Homologues dans :
 - Lidar aérien,
 - Lidar sol (MNS)
 - Orthophoto

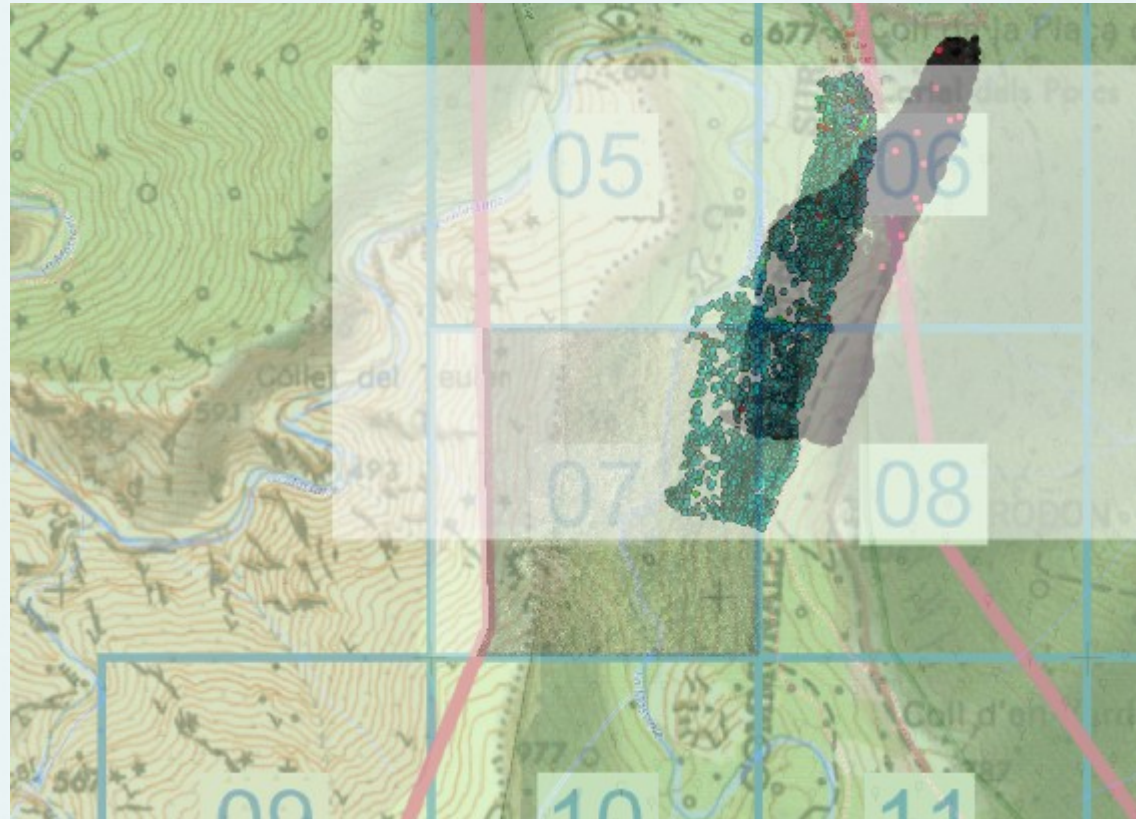


Extraction Zone Intérêt dans QGIS



Ci contre en RFG 93 :

- Fond de carte OSM,
- Zonage Sintegra
- Marron + pts rouges relevés manuels RNN
- Points Verts proposition Exametrics 03/19
- Pt rouge dans points verts => refuge Colomates



La bande rouge dans le zonage Sintegra, donne les limites des relevés LIDAR aériens pris en compte dans les semis et MNT/S/C Sintegra. Le décalage des données initiales RNN est lié à un calage sur un référentiel IGN obsolète vu l'antériorité des premiers relevés RNN.





Tuiles réduites MNT

- Hypothèse Nuage LIDAR E*¹ repositionné et recalé correctement en RFG93
- Carroyage pour réduire les zones de traitement
 - 100 m cf. relevés Lamassane sur nuage E*¹
 - 500 m cf. relevés Sintegra sur nuage E*¹
- Calcul de cellule moyenne MNT de 25 m² ou 4 m² (5m x 5m voire 2m x 2m) à partir du lot de nuages MNT S*², ne présente pas d'intérêt interpolation imprécise (issu travaux voir écran suivant).
- Mise tous les pieds d'arbres à Z=0, hypothèse de planéité de la cellule de 25 ou 4 m², de toutes manières la déviation standard est accessible pour chaque cellule (voir plus loin 56 et + remplacement : MNT extraite des mesures EXAMETRICS)

Fournitures : *¹ => EXAMETRICS, *² => Sintegra



Cellule MNT réduite



- Construction depuis les fichiers MNT Sintegra
- Reconstruction plus précise avec données EXAMETRICS (cf. Ecrans 56 et +)
- Zone d'intérêt Sintegra (5/6/7/8)
- Stockage en fichiers en mode binaire 2m x 2m ou 5m x 5m
- Stockage MNT cellulaire dans structure associative Map non ordonnée, accès direct par index
- Index (string concat xmin-ymin (modulo taille))
- Un index => une cellule (5m x 5m ou 2m x 2m) qui contient :
 - Hauteur moyenne $\text{MeanZ} = Z_{\text{réduit}}$
 - Nombre de points dans la cellule nbPoints
 - StdDev écart-type de Z sur la cellule





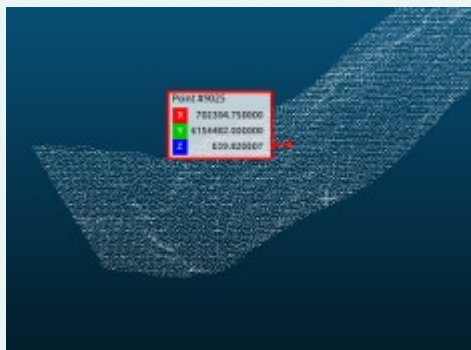
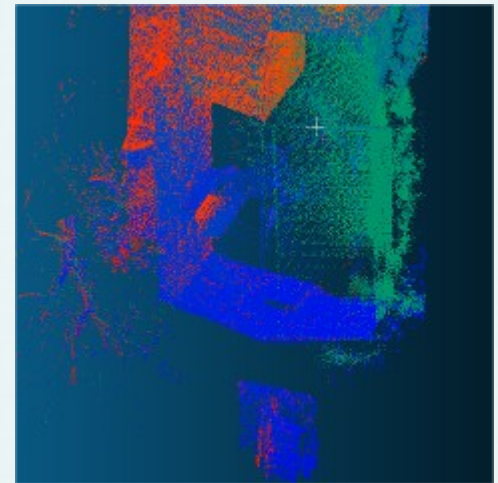
Règles de recalage

- Le seul amer connu lors du premier dépouillement est le refuge des **Colomates** :
 - Visible et géolocalisable sur les images TIFF S^{*2} (Bâtiment + table orientation SUD),
 - Visible et géolocalisable dans nuage global E^{*1}
 - Visible et géolocalisable dans le nuage clôture E^{*1}
 - Ceci permet un estimatif grossier de la transformation de Helmert (Matrice 4 x 4) pour recaler les nuages (Global & Clôture) E^{*1} en RFG93

Fournitures : $*^1 \Rightarrow$ EXAMETRICS, $*^2 \Rightarrow$ Sintegra



Données de recalage



Dans l'ordre : TIFF, Gros Nuage, Clôture, et fragment de MNS (50m x 50m).

La construction de la transformation de Helmert Matrice 4x4 est faite à partir de ces 4 points
Empreinte au sol de 12 m² pour recaler une empreinte de 9 ha source majeure d'erreurs notamment en 3D.



Données prises en compte RFG 93



Extraction Image TIFF + MNS
Cible à atteindre (C)

C

Extraction Nuage global
(S) Source à transférer + (T) Translation

S

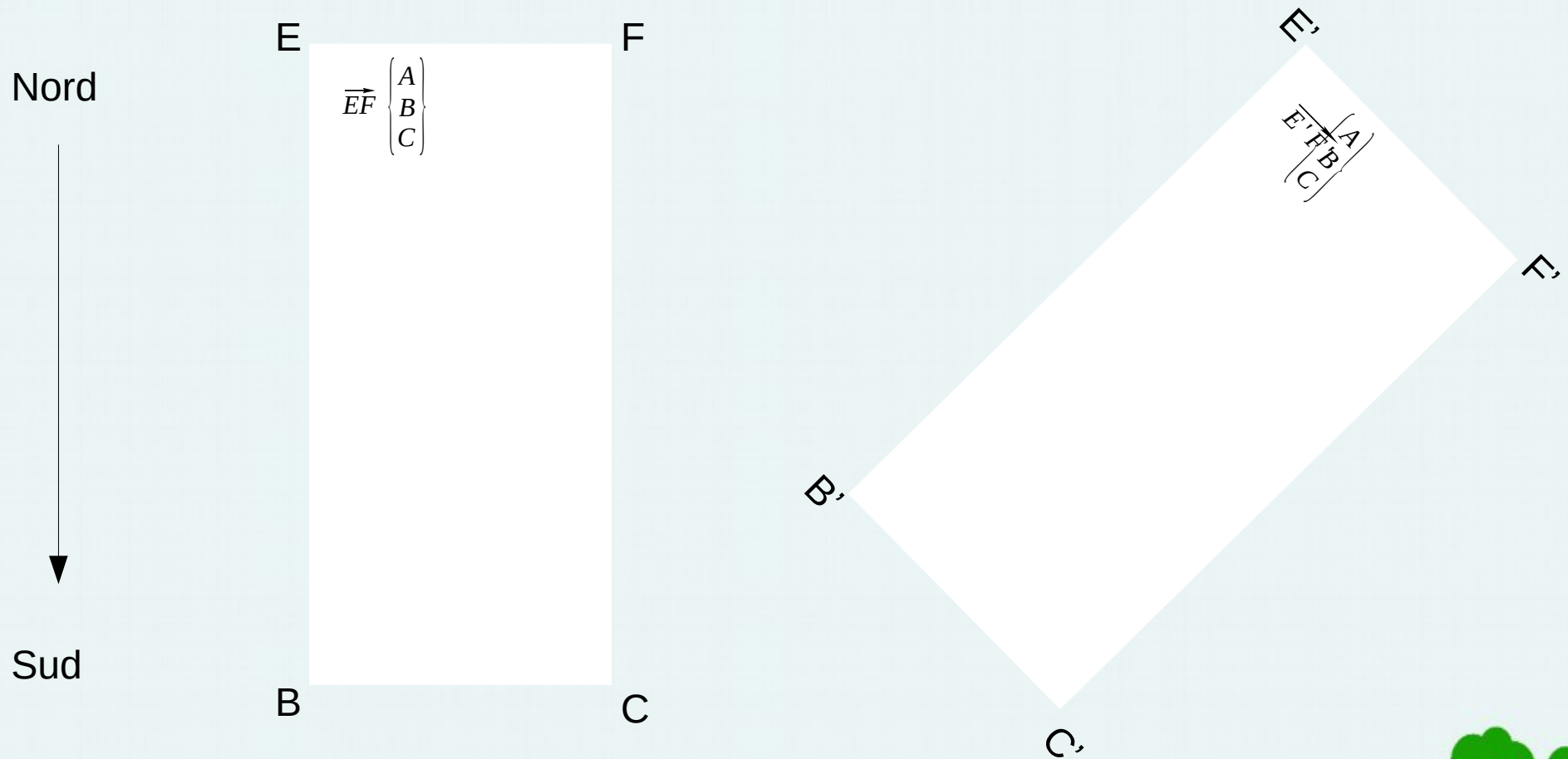
T

E	X : 702 395.60 Y : 6 154 482.58 Z : 644.09	X : 320.47 Y : 449.76 Z : -57.01	702 075.13 6 154 032.82 701.1
B	X : 702 395.60 Y : 6 154 482.58 Z : 644.14	X : 315.88 Y : 453.65 Z : -54.86	702 079.72 6 154 029 .93 699.0
C	X : 702 395.60 Y : 6 154 482.58 Z : 642.62	X : 314.03 Y : 451.68 Z : -57.83	702 081.57 6 154 030 .90 700.54
F	X : 702 395.60 Y : 6 154 482.58 Z : 642.909	X : 318.68 Y : 447.78 Z : -54.92	702 076.92 6 154 034 .80 697.829

Vu le nombre de chiffre significatif la translation en flottant est une source d'erreurs, nécessité de séparer cette opération en mode entier



Géométrie basique



Protocole de transformation (1/5)



Qu'il s'agisse de transformations rigides ou non rigides de nuage de points, l'essentiel des calculs porte sur des opérations locales et globales de rotation, de déformations hétérogènes de coordonnées, voire à contrario de mise à l'échelle uniforme. Des précautions sont à prendre pour éviter les erreurs de calcul (nombre de variables significative en flottant)

Dans tous les cas il s'agit de translation, de rotation au sein d'un repère, autour d'un axe arbitraire, ou de parallélisation de plan.

Dans ce qui suit nous présentons les outils mathématiques de base vecteurs, matrices, quaternions permettant de conduire ces opérations, afin d'obtenir des calages, recalages et superpositions géolocalisées





Protocole de transformation (2/5)

- Deux triangles identiques ($P_1, \vec{N}_1$ et $P_2, \vec{N}_2$) P_2 cible, P_1 point à transformer
 - Paralléliser $\vec{N}_1$ et $\vec{N}_2$ et les orienter à l'identique,
 - Transformer nuage P_1 , avec données orientation,
 - Translater la droite support des normales pour superposition et Transformer nuage P_1 , avec données translation,
 - Assurer la rotation du nuage P_1 , avec données rotation autour de N_1 faisant coïncider les trois sommets,





Parallélisation de 2 normales (3/5)

- Normales ex. 2 vecteurs unitaires $\vec{A}, \vec{B}$ (angle Θ)

- *Produit vectoriel* $\vec{A} \times \vec{B} = \vec{V}$

Module =1 pour tous ($\vec{A}, \vec{B}$)

$$\vec{V} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$$

- $s = ||V|| \Rightarrow \sinus \Theta, C = \text{produit scalaire } \vec{A} \cdot \vec{B} = \cos(\Theta)$

- *Matrice [R] parallélisant* $\vec{A}, \vec{B} \Rightarrow [R] = [I] + [V] + [V]^2 \frac{1-C}{s^2}$ avec $\frac{1-C}{s^2} = K$

- [V] matrice assymétrique de $\vec{V}$ $[V] = \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}$

Matrice rotation 3 x 3 $\Rightarrow [R] = \begin{bmatrix} 1 + K(-v_z^2 - v_y^2) & Kv_x v_y - v_z & Kv_z v_x + v_y \\ Kv_x v_y + v_z & 1 + K(-v_x^2 - v_z^2) & Kv_x v_z - v_y \\ Kv_x v_z - v_y & Kv_y v_z + v_x & 1 + K(-v_x^2 - v_y^2) \end{bmatrix}$





Evaluation du quaternion unitaire de rotation (4/5)

Le quaternion unitaire sera porté en direction par le vecteur de référence $\vec{V}_{ref}$ dans le nuage destination. Pour sa construction il faudra se baser sur son vecteur homologue $\vec{V}_{hom}$ dans le nuage à transformer.

La rotation Θ est l'angle entre le vecteur de référence $\vec{V}_{ref}$ et son homologue les deux étant définis dans le même référentiel.

Les deux vecteurs étant définis par leurs coordonnées, afin de rester dans un trièdre direct, Θ est confirmé sans ambiguïté par les calculs des produits scalaires et vectoriels tels que :

$$\Theta = \arccos \frac{\vec{V}_{ref} \cdot \vec{V}_{hom}}{|\vec{V}_{ref}| |\vec{V}_{hom}|} = \arcsin \frac{|\vec{V}_{ref} \times \vec{V}_{hom}|}{|\vec{V}_{ref}| |\vec{V}_{hom}|}$$





Matrice issue du Quaternion de rotation autour d'un vecteur unitaire (5/5)

$$q = q_0 + \vec{Q}_v$$

$$\vec{Q}_v = i q_1 + j q_2 + k q_3$$

$$q = q_0 + Q_v = q_0 + i q_1 + j q_2 + k q_3 = \cos \Theta / 2 + \vec{V} \sin \Theta / 2$$

Le quaternion q définit le vecteur unitaire $\vec{V}$ autour duquel se fait la rotation Θ d'un point P

La rotation est effectuée par $q P q^{-1} = [M][\vec{P}]$ avec $\rightarrow [\vec{P}] = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}$

$$[\vec{Q}_v] = \begin{bmatrix} Q_{vx} = \frac{q_1}{\sin \Theta / 2} \\ Q_{vy} = \frac{q_2}{\sin \Theta / 2} \\ Q_{vz} = \frac{q_3}{\sin \Theta / 2} \end{bmatrix}$$

Peut être mis sous forme matricielle

Matrice de rotation 3x3 $\Rightarrow [M] = \begin{bmatrix} 1 - 2q_2^2 - 2q_3^2 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & 1 - 2q_1^2 - 2q_3^2 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & 1 - 2q_1^2 - 2q_2^2 \end{bmatrix}$





Rotation angle d'Euler et quaternions

Un rotation d'un angle Θ autour d'un vecteur unitaire $\vec{u} = u_x \vec{i}, u_y \vec{j}, u_z \vec{k}$ peut être représenté par un quaternion en extension de la formule d'Euler, les composantes du vecteur unitaire, axe de rotation sont définies dans le référentiel initial :

$$q = e^{\frac{\Theta}{2}(u_x \vec{i}, u_y \vec{j}, u_z \vec{k})} = \cos(\Theta/2) + (u_x \vec{i}, u_y \vec{j}, u_z \vec{k}) \sin(\Theta/2)$$

La rotation d'un vecteur $\vec{V}$ obéit à l'équation sandwich suivante : $q \vec{V} q^{-1}$





Evaluation des matrices de transformation

Source données page 29 (1/5)

- On construit les normales sources et destinations à partir des vecteurs homologues $\vec{EB}$ et $\vec{EC}$ en source et destination
- Par projection sur les vecteurs unitaires $\vec{i} \vec{j} \vec{k}$ on estime en respectant l'orientation du référentiel l'angle des composantes $\vec{N}_{source}$ et $\vec{N}_{cible}$
- Par exemple angle par rapport à OZ pour $\vec{N}_{source}$: $\Theta_{source-OZ} = \frac{x_s}{\sqrt{x_s^2 + y_s^2}}$
- Les autres angles sont obtenus par permutation circulaire en respectant la règle du trièdre direct, idem pour $\vec{N}_{cible}$
- Ceci donne pour chaque axe du référentiel, exemple pour OZ
Rotation autour de OZ : $\Theta_{cible-OZ} - \Theta_{source-OZ}$
- Les rotations entre les 3 axes étant connues, ceci permet de construire la matrice de rotation (3x3) à laquelle l'ajout du vecteur translation $\mathbf{T}$ permet de finaliser la matrice 4 x 4 d'Helmert.
- La rotation se fait dans un référentiel parallèle au référentiel de mesure mais avec son origine au point E de l'image source (Donc translation $\vec{T}_{ref}$ à prendre en compte dans les coordonnées de points).



Evaluation des matrices de transformation

Source données page 29 (2/5)



Etape 1 : Matrice de rotation et de translation superposition des normales et du point E de la source avec son homologue en destination

La matrice de rotation est construite par le **produit** des matrices 3 x 3 des rotations indépendantes par rapport aux axes $\vec{OX}, \vec{OY}, \vec{OZ}$ du référentiel.
La commutativité est sans influence.

Vecteur Translation $\vec{T} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix}$

$$[H] = \begin{bmatrix} \begin{bmatrix} \text{Matrice} \\ 3 \times 3 \\ \text{de Rotation} \end{bmatrix} & \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \\ \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} & \begin{bmatrix} 1 \end{bmatrix} \end{bmatrix}$$

Matrice 4 x 4 de transformation de Helmert
1ère transformation :

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} = [H] \begin{bmatrix} x_{init} \\ y_{init} \\ z_{init} \\ 1 \end{bmatrix}$$



Evaluation des matrices de transformation

Source données page 29 (3/5)



Matrice de rotation autour de la normale unitaire passant par E : $\overrightarrow{N}_{cible-unitaire}$

L'ensemble des matrices de transformation sont des matrices orthonormées qui conservent les distances et les angles. Donc pour les superpositions il suffit d'évaluer l'angle α entre $\overrightarrow{EB}$ dans l'image TIFF et son homologue dans le nuage de points tous deux dans le même référentiel.

$\overrightarrow{N}_{cible-unitaire} = \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix}$ si une rotation α est nécessaire pour faire coïncider les homologues B ou C en appelant $c = \cos(\alpha)$ et $s = \sin(\alpha)$

$$\left[Rot_{\overrightarrow{N}_{cible-en E}} \right] \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} c + (1-c)N_x^2 & (1-c)N_xN_y - sN_z & (1-c)N_xN_z - sN_y \\ (1-c)N_xN_y + sN_z & c + (1-c)N_y^2 & (1-c)N_yN_z - sN_x \\ (1-c)N_xN_z - sN_y & (1-c)N_zN_y + sN_x & c + (1-c)N_z^2 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

Ceci achève le processus de transformation et de recalage d'un nuage de points en utilisant des éléments homologues entre le nuage source et la zone cible, cette zone cible peut être un autre nuage ou simplement des amers, mais dans les deux cas les points homologues géométriques doivent être identifiés



Evaluation des matrices de transformation

Source données page 29 (4/5)



- Après ces opérations les plans (pas les triangles, car rotation encore nécessaire dans le plan) des triangles homologues ont un point commun E, ou passent les deux normales unitaires qui se confondent
- Maintenant il faut procéder à la stricte superposition des triangles, ceci est obtenu par une rotation autour du vecteur unitaire normal maintenant commun et passant par E, ceci est obtenu par une rotation autour de $\vec{N}_{cible}$ qui se confond maintenant avec $\vec{N}_{source}$

tel que : $\vec{N}_{cible-unitaire} = \begin{pmatrix} N_x \\ N_y \\ N_z \end{pmatrix}$ si une rotation α est nécessaire pour faire coïncider

les homologues B ou C en appelant $c = \cos(\alpha)$ et $s = \sin(\alpha)$, cette dernière rotation est obtenue avec la matrice de rotation qui suit.



Evaluation des matrices de transformation

Source données page 29 (5/5)



Seconde transformation rotation autour de la normale unitaire passant par E
Ce qui donne la matrice globale de transformation [Trsf]

$$\begin{bmatrix} x_{final} \\ y_{final} \\ z_{final} \\ 1 \end{bmatrix} = \begin{bmatrix} Rot_{\overrightarrow{N_{cible-en E}}} \\ [0 & 0 & 0] \\ [1] \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \\ 1 \end{bmatrix} = \begin{bmatrix} Rot_{\overrightarrow{N_{cible-en E}}} \\ [0 & 0 & 0] \\ [1] \end{bmatrix} [H] \begin{bmatrix} x_{init} \\ y_{init} \\ z_{init} \\ 1 \end{bmatrix} = [Trsf] \begin{bmatrix} x_{init} \\ y_{init} \\ z_{init} \\ 1 \end{bmatrix}$$

$$[Trsf] = \begin{bmatrix} Rot_{\overrightarrow{N_{cible-en E}}} \\ [0 & 0 & 0] \\ [1] \end{bmatrix} \begin{bmatrix} Matrice \\ 3 \times 3 \\ de Rotation \\ [0 & 0 & 0] \\ [1] \end{bmatrix} \begin{bmatrix} T_x \\ T_y \\ T_z \\ 1 \end{bmatrix}$$





Evaluation des matrices de transformation

Source données page 29 (3/3)

$\vec{A}$ et $\vec{B}$ deux vecteurs Orthonormés libres en 3 D avec angle Θ

$$G = \begin{bmatrix} \cos(\Theta) & -\sin(\Theta) & 0 \\ \sin(\Theta) & \cos(\Theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad G = \begin{bmatrix} A \cdot B & -\|A \times B\| & 0 \\ \|A \times B\| & A \cdot B & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ci-devant deux matrices de rotation de $A \searrow B$ dans le référentiel suivant :

- Projection de A sur B $\mathbf{a}_1 = a_1 \hat{\mathbf{b}} = (|\mathbf{a}| \cos \theta) \hat{\mathbf{b}}$

$$u = (A \cdot B) \frac{A}{\|(A \cdot B) A\|} = A$$
- Réjection de B sur A $\mathbf{a}_2 = \mathbf{a} - (|\mathbf{a}| \cos \theta) \hat{\mathbf{b}}$. (perp. à B)

$$v = \frac{B - (A \cdot B) A}{\|B - (A \cdot B) A\|}$$
- La normale à B et A

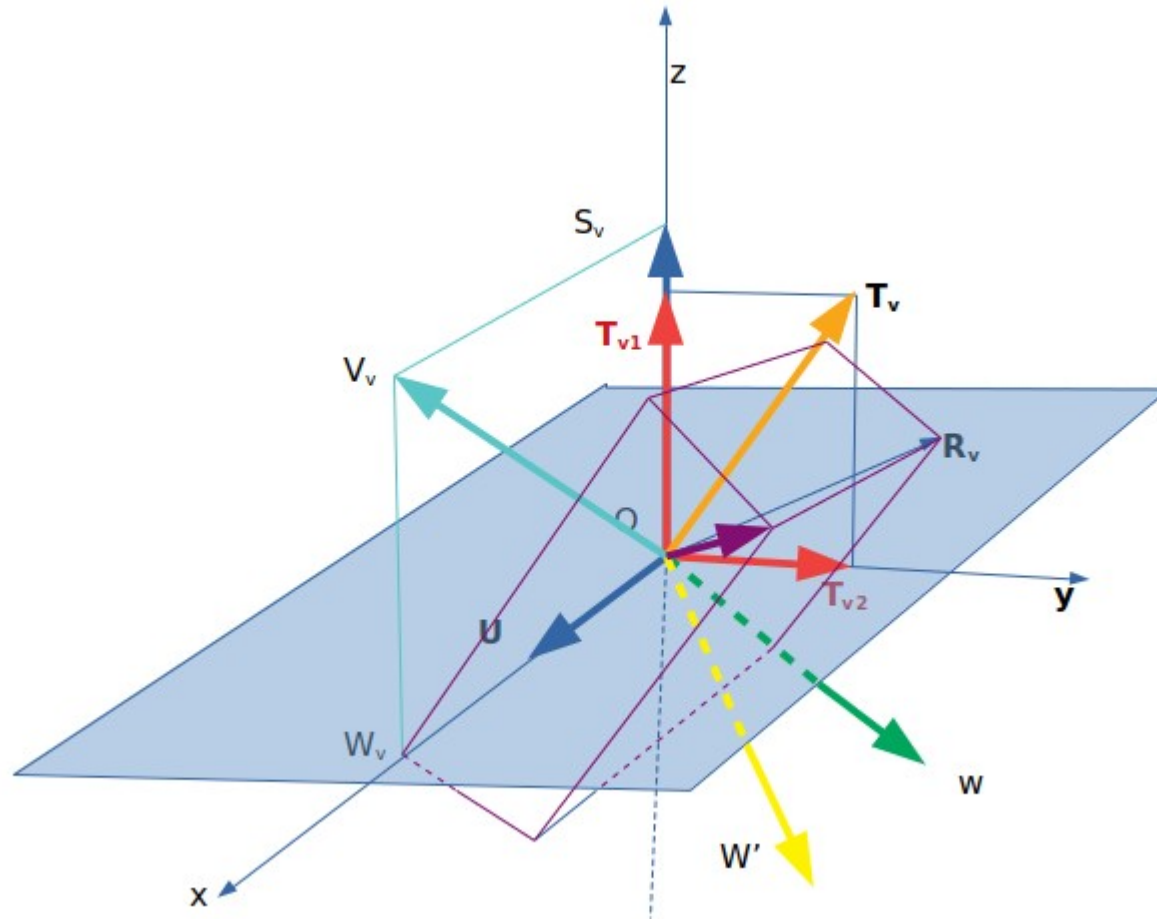
$$w = A \times B$$

La matrice de changement de base est donc :

$$F = [u \quad v \quad w]^{-1} = \begin{bmatrix} A & \frac{B - (A \cdot B) A}{\|B - (A \cdot B) A\|} & B \times A \end{bmatrix}^{-1}$$



Rotation par quaternion – Géométrie (1)





Rotation par quaternion – Géométrie (2)

U un vecteur quaternion unitaire, Sv un vecteur quaternion

Rappel, deux quaternion **p** et **q** => leurs propriétés :

$$q = q_0 + iq_1 + jq_2 + kq_3 = q_0 + V_q \quad p = p_0 + ip_1 + jp_2 + kp_3 = q_0 + V_p$$

$$\text{Le produit} \Rightarrow qp = q_0p_0 - V_q \cdot V_p + q_0V_p + p_0V_q + V_q \times V_p$$

$$\text{si } p \text{ et } q \text{ orthogonaux et pures vecteurs} \Rightarrow qp = q_0p_0 - V_q \cdot V_p + q_0V_p + p_0V_q + V_q \times V_p, \text{ où : } \{.\} \text{Produit scalaire, } \{\times\} \text{Produit vectoriel.}$$

Référentiels orthonormés directs : xoyz, $UOT_{v_1}T_{v_2}$, UOT_VW

U étant un quaternion unitaire =>

=> T_v est une rotation de S_v autour de l'axe portant U, c'est à dire OX

Considérons maintenant la rotation suivante R_v :



Rotation par quaternion – Géométrie (2)



- A reprendre ! En levant toutes les ambiguïtés de rotation, erreur pouvant être due au non respect de trièdre direct (toujours calculer l'angle entre deux vecteurs en y introduisant la normale d'observation
bibliothèque vg => travailler dans le domaine $[-180^\circ, +180]$ pour exemple :
 - `i=np.array([1.0,0.0, 0.0])`
 - `j=np.array([0.0,1.0, 0.0])`
 - `k=np.array([0.0,0.0, 1.0])`
 - `angle=vg.signed_angle(j, i, look=k)`
- Sera nécessaire pour des interpolations spatiales de correction
- A l'instar de ce qui est fait en interpolation temporelle en simulation de mouvement





QUALITE FINALE (1/3)

La qualité de la transformation est totalement liée à la qualité de la mesure initiale.

Une transformation exacte sur des données fausses Donne des résultats faux, sans intérêt.

Primeur du protocole et de la calibration de la mesure initiale.

La transformation présentée, donne le principe, elle n'a pas prétention à l'exactitude, en effet les surfaces manipulées sont trop faibles par rapport à l'empreinte du nuage, donc erreurs de calcul.

En outre il est nécessaire d'en manipuler plusieurs surfaces. D'où l'importance de l'identification ou de la mise en place préalable d'amers géoréférencés dont les points homologues sont aisément identifiables dans le nuage de points. Ceci donne des géométries linéaires dont l'usage est, aussi, une alternative déterministe aux algorithmes ICP en agrégation (« registration ») dont la convergence est peu assurée et les calculs erronés du fait des nombreuses itérations en flottant.





QUALITE FINALE (2/3)

Que le calcul se base sur une rotation par rapport à un axe qqconque ou sur une évaluation de quaternion
Il y a tout intérêt à évaluer plusieurs matrices
quitte à prendre la moyenne, avec les quaternions,
une interpolation spatiale est envisageable.

Ceci est possible dans l'hypothèse d'une multitude
d'identification de vecteurs ou points de référence dans le
Le référentiel de destination avec leur homologues dans
le nuage à transformer défini dans le même référentiel.

Des précautions sont à prendre en matière de calcul.
Travailler dans des repères réduits afin de réduire le
nombre de chiffres significatifs car précision $> 10^{-6}$



QUALITE FINALE (3/3)



Précautions particulières aux calculs de CUDA, qui travailler en double mais afin de Limiter les propagation d'erreur des flottants, et autres précautions , Quelques références :

- <https://developer.download.nvidia.com/assets/cuda/files/reduction.pdf>
- https://en.wikipedia.org/wiki/Kahan_summation_algorithm
- <http://www.iro.umontreal.ca/~mignotte/IFT2425/Documents/AccrateSummationMethods.pdf>
John Michael McNamee, "A Comparison Of Methods For Accurate Summation." ACM SIGSAM Bulletin, Vol. 38, No. 1, March 2004, pp. 1-7
- <http://www.ti3.tu-harburg.de/paper/rump/Ru08b.pdf>
Siegfried M. Rump, "Ultimately Fast Accurate Summation." SIAM J. Sci. Comput., Vol. 31, No. 5, Sep. 2009, pp. 3466–3502
- https://www.jstage.jst.go.jp/article/nolta/1/1/1_1_2/_pdf
Siegfried Rump, Takeshi Ogita, and Shinichi Oichi, "Fast high precision summation." Nonlinear Theory and Its Applications, IEICE 1.1 (2010): 2-24
- https://people.eecs.berkeley.edu/~hdnguyen/public/papers/ARITH21_Fast_Sum.pdf
- James Demmel and Hong Diep Nguyen, "Fast Reproducible Floating-Point Summation." In: 21st IEEE Symposium on Computer Arithmetic (ARITH21), April 2013, pp. 163-172
- https://www.math.uu.se/digitalAssets/562/c_562622-l_1-k_nehmeier_slides.pdf
- Alexander Dallmann, Fabian Jörg, Marco Nehmeier and Jürgen Wolff von Gudenberg, "Correctly Rounded Dot Product in CUDA. (2014)
- <https://hal.archives-ouvertes.fr/hal-00949355v1/document>
Sylvain Collange, David Defour, Stef Graillat, Roman Iakymchuk, "Full-Speed Deterministic Bit-Accurate Parallel Floating-Point Summation on Multi- and Many-Core Architectures" (Sep. 2015)
- Technique antérieure => <https://hal.inria.fr/inria-00074262/document>

Dans le cas présent (RFG93) nous sommes confrontés à des translations nécessitant des précision à 10^{-7} nous l'extrayons de la matrice de transformation et nous achevons le cacul en entier avec la bibliothèque GNU BigNum GMP pour la translation



Précautions en phase de relevé TLS avec centrale inertielle (I.MU) [1/2]



- Se munir d'une carte IGN ou équivalent et planifier le parcours physique en quadrillant les zones,
- A chaque relevé, recalé l'IMU, TLS sur TRIPOD :
 - s'assurer de l'horizontalité du TLS lors du recalage
 - Orienter le TLS axe Nord/Sud pointé vers le nord,
 - Relever sa position GPS initiale (si nécessaire trouver le lieu de captation satellitaire, un GPS type Etrax Garmin fait l'affaire dans un premier temps, le luxe du différentiel non nécessaire),
- Identifier tout amer (Naturel, par photo, réflecteur LIDAR, posé au sol avec localisation GPS obligatoire) , **nécessairement** détectable dans le nuage à venir,
- Toute localisation GPS d'arbre est à faire au pied de celui-ci et au sud de l'arbre pour préserver les cohérences de mesures.



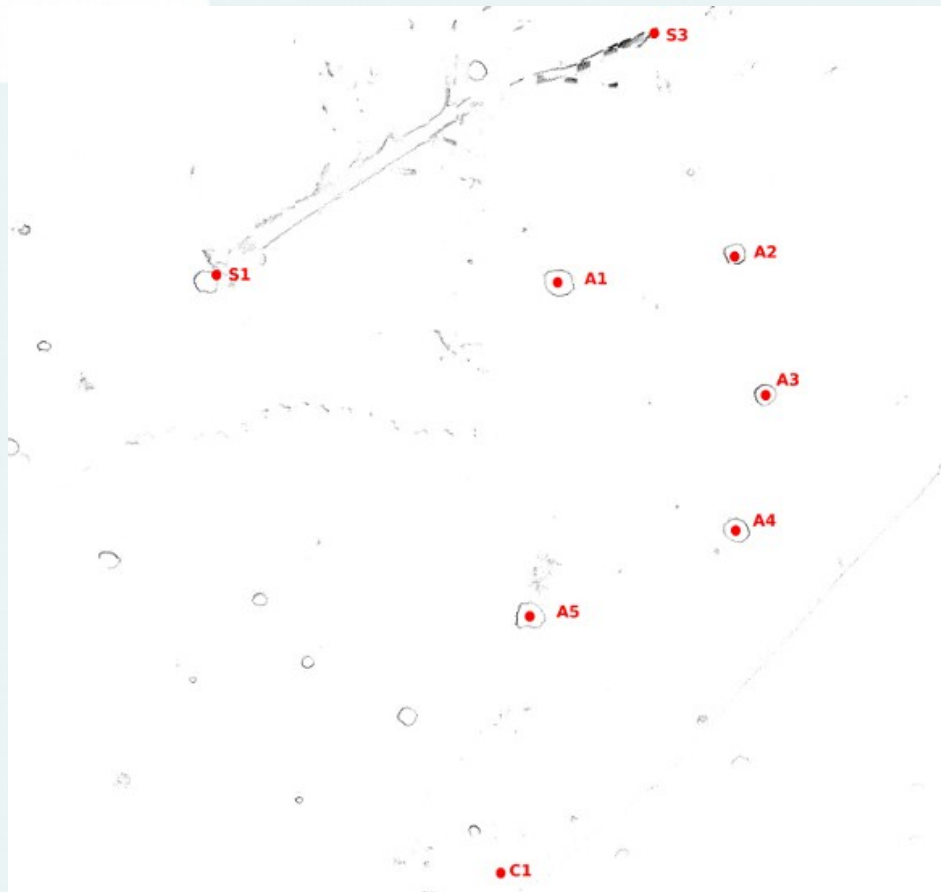
Précautions en phase de relevé TLS avec centrale inertielle (I.MU) [2/2]



- Si agrégation à venir chevaucher 10 à 20 % du relevé antérieur avec points homologues (et amers) communs entre relevé N et N+1,
- Limiter le temps d'acquisition du fait de la dérive de la centrale inertielle qui doit être connue a priori,
- Autant que possible relever point départ et final GPS,
- Bien référencer le relevé et ses données associées (toutes informations pertinentes pour le positionnement),
- Créer une gille de remontage des nuages sur le plan initial en référençant parfaitement les zones,
- Chaque relevé LIDAR de zone doit être accompagné, **au moins**, de trois points de relevés GPS, de manière à construire des vecteurs aussi grands que possibles, non colinéaires et orthogonaux si possible pour 2 d'entr' eux



Exemple de relevé d'amers sur le terrain



Dans le cas présent, section de 10 cm du nuage de point à 60 cm, du niveau Zéro, on a un amer spécifique (arbre au sol) Points S1 et S3. A proximité on identifie:

- un point de clôture C1
- 5 arbres A1 à A5.

Pour chacun de ces points, un relevé géodésique (X, Y Z) nous donne les éléments de calcul de la matrice de transformation. On dispose de 8 points de mesures donc 7! vecteurs, et 6 ! couple de vecteurs pour évaluer la matrice de transformation (moyenne)
Voir écran 90+

La validation des relevés (amers) se fait en comparant , respectivement, la longueur des vecteurs obtenus et l'angle entre couples de vecteurs, avec les attributs similaires sur les homologues du nuage de point **en respect de la marge d'erreur acceptée.**



ACCELERATION DES CALCULS



Le nuage global E^{*1} présente 10^9 points, toutes opérations de calcul pour une transformation (rotation/translation matrice 4×4) représente par point 16 multiplications et 12 additions flottantes par points.

La même problématique se présente pour les filtrages et les opérations de convolution/corrélation

A ces opérations il faut ajouter les échanges mémoires, tout ceci doit être optimisé afin de réduire le coût du calcul.

Ces opérations sont reconduites lors du moindre déplacement du nuage en phase de calcul.

Dans les phases exploratoires il est fortement recommandé de travailler sur des nuages de petites tailles, d'où nécessité de carroyages divers.

Les gros nuages sont un obstacle pour le temps de calcul et la précision





Technique d'agrégation recalage retenue [1]

- Dans le cas présent, le recalage **ne concerne pas l'agrégation de nuages de points** mais **le strict positionnement d'un nuage de points dans un référentiel géolocalisé**, ici RFG93, (donc translation et rotation 3D).
- Classiquement les recalage ou agrégations sont faites par l'algorithme ICP dont la grande diversité des variantes s'explique par la difficulté de la convergence et l'imprécision des résultats voir => travaux (Marcel Bariou et Mahdi Smida Eté 2018), on renonce, en outre c'est un sujet collatéral (car agrégation accroît les erreurs).
- On retient une approche déterministe pure avec les groupes de rotations on affinera plus tard si nécessaire, avec retour aux quaternions, notamment si besoin d'interpolations spatiales.
- **Nuage de référence** On dispose ou on construit par calcul un nuage calibré contenant au moins trois points géoréférencés.
- **Nuage source** ou nuage à recaler (on supprime les zones douteuses et on refait les mesures correspondantes en respectant les protocoles) sur le nuage de référence.
- Le **nuage source** (possiblement déformé ou mal géolocalisé) possède des points de référence identifiables dont les géolocalisations sont connues. Idem pour leurs points homologues dans le nuage de référence.





Technique d'agrégation recalage retenue [2]

Construction de la matrice de transformation

Les 3 points sont des points homologues dans les deux nuages :

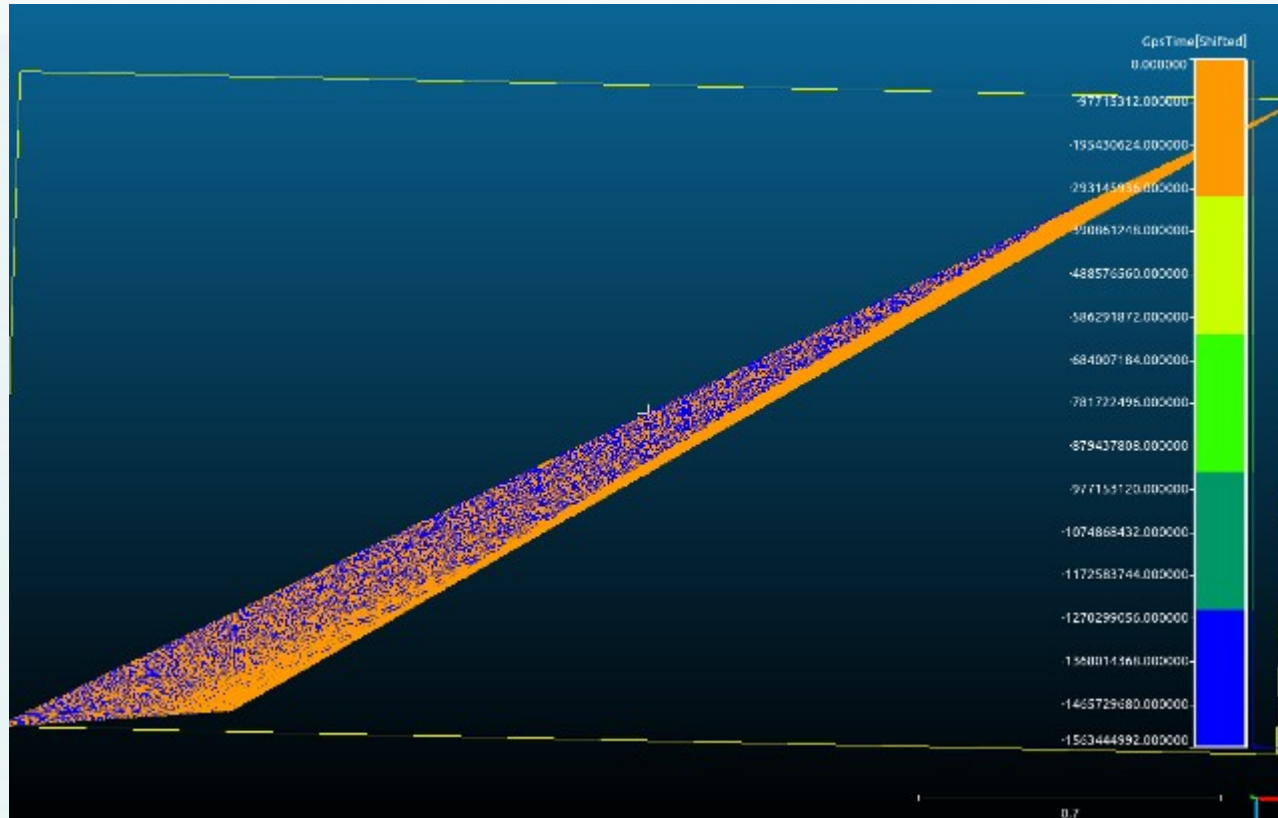
- Construire des triangles orientés (respect impératif),
- Privilégier la possibilité de construire un triangle rectangle ou presque, pas d'angles $> 90^\circ$ si possible, les côtés les plus longs possibles,
- Les triangles source et référence sont strictement égaux normalement (parfois mise à l'échelle nécessaire) mais des écarts peuvent subsister :
 - anomalie de mesure, - dérive de central inertielle, - etc. mais bien identifier les associations de points homologues
- Les deux nuages sont dans le même référentiel de là :
 - Translater $\Rightarrow$ **Mat0** entre deux point homologues confondu en **M1**
 - Paralléliser les normales source et référence (1 ère rotation autour du point **M1**, portant l'axe de rotation perpendiculaire au plan des deux normales) $\Rightarrow$ **Mat1**
 - Correction de rotation (2 ème totation) entre deux vecteurs homologues $\Rightarrow$ **Mat2** on a coïncidence
- La matrice globale de rotation/translation est MatTRot

$$\mathbf{MatTrot} = [\mathbf{Mat0}][\mathbf{Mat1}][\mathbf{Mat2}]$$

les matrices élémentaires étant calculées après chaque transformation



Technique d'agrégation recalage retenue [3]



Superposition finale de deux triangles (bleu et orange) , a priori homologues mais non identiques, source mal mesurée, distance > 700 km (contexte d'imprécision : manipulation de vecteurs de taille de 6 m avec des translations $> 700 \text{ km} \cdot 10^5$), c'est le type même de configuration de calcul à éviter, en flottant, car propice à la propagation des erreurs.





Technique d'agrégation recalage retenue [4]

RECAPITULATIF

Pour un nuage source de point LIDAR, si à la visualisation, il y a inadéquation entre les relevés terrain des amers et leur position dans le nuage visualisé :

- La matrice de transformation doit être calculée,
- Relever les points amers visualisés d'un triangle du nuage source,
- Introduire les valeurs attendues triangle du nuage référence
- Lancer le calcul de la transformation en validant les résultats obtenus
- Récupérer la matrice de transformation obtenue
- Transférer la matrice de transformation dans le transformateur C_CUDA
- Lancer la transformation C_CUDA (**Si Amers corrects initialement on part d'ici**)
- Récupérer le nuage correctement positionné
- Indexer le nuage pour obtenir son empreinte au sol (Couche SIG éventuelle)
- Retirer le relief avec le tuilage réduit MNT (à adapter selon la précision obtenue)
- Lancer l'identification de positionnement et de calcul attributaire des arbres
- Récupérer la couche SIG correspondante avec les données attendues



Extraction MNT du nuage EXAMETRICS



- **Le lidar mobile terrestre offre l'avantage de fournir de nombreux points au sol**
- Le relief est très accidenté donc le seuillage global en Z impossible
- Les MNT Sintegra souffrent d'une résolution insuffisante (plusieurs raisons possibles : résolution de mesures et règles d'interpolation)
- Tuilage (carroyage) nécessaire avec un encombrement 2m x 2m :
 - Respecte largement l'espacement des arbres
 - Accepte des arbres de 6 m de diamètre (1m de rayon)
- Une extraction statistique du modèle de terrain est possible
- Substitution à un vol aérien avec lidar multiécho classifiant (économie importante)
- Les étapes :
 - Établir l'encombrement précis de la zone (MinMax X,Y)
 - Carroyage (X, Y) avec attribut par cellule sur le Z :
 - Min, max,
 - Nb de points dans la cellule
 - Valeur moyenne Z (meanZ)
 - Ecart type de la mesure
 - Filtrage paramétré
- Génération d'un fichier texte rassemblant les attributs par cellule
- Pour visualisation semis régulier de point à hauteur meanZ par cellule et génération d'un *.las
- La qualité et la précision d'un modèle de terrain sont fondamentales pour le dimensionnement d'un arbre (hauteur Z)





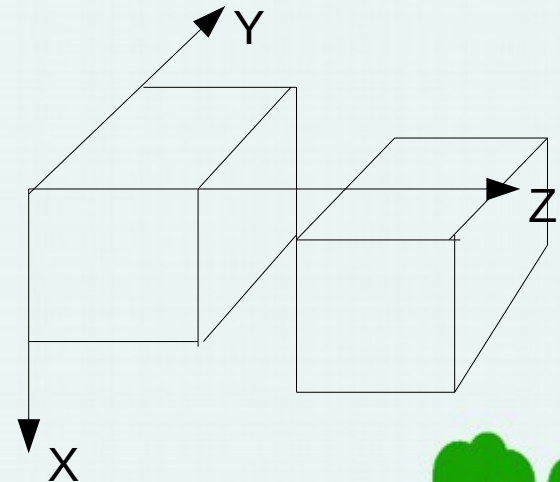
1 : Encombrement et règle de carroyage

- **Première lecture disque** du fichier et ***traitement à la volée***
- Établir l'encombrement précis de la zone (MinMax X,Y)
- Génération des règles carroyages, principe permettant un carroyage 2D (cellule) et 3D (voxel)
- Pas de parcours récursif, tout est en accès direct en mode aléatoire, adressage par un index construit en concaténant ses coordonnées (X, Y) [cellule] et Z [Voxel] modulo le pas de carroyage
- l'identification de la présence ou non, du point, dans la zone carroyée par son index déduit de ses coordonnées comme indiqué.

$$X_{\text{modulo}} = [(\text{int})(X_{\text{réel}} / \text{fpas}_x)] * \text{lpas}_x$$

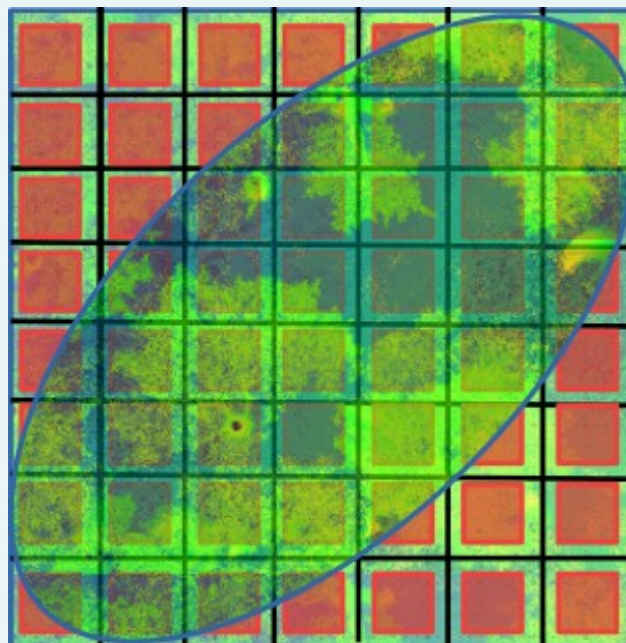
$$\text{Index} = X_{\text{modulo}} ! Y_{\text{modulo}} ! Z_{\text{modulo}}$$

Ainsi dans un triède direct on peut rassembler les points LIDAR dans une cellule(2D) ou un voxel(3D) et avoir un accès direct et rapide de manière aléatoire **sans octree**.





1 : Encombrement et règle de carroyage



Le carroyage s'effectue sur l'ensemble de l'encombrement du nuage de points traité, modèle stylisé ci-dessus sur une petite zone.



2 : Carroyage, stats cellule

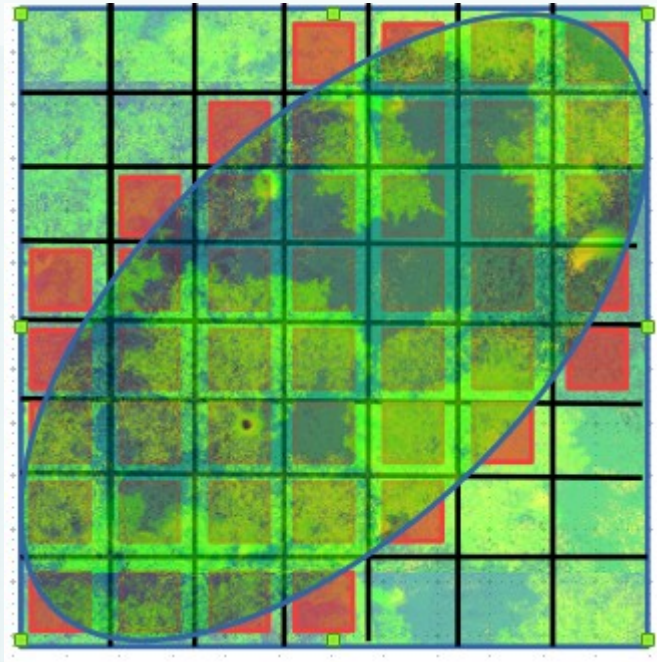


- **Deuxième lecture du fichier** et traitement à la volée
- Carroyage (X, Y), indexation avec attributs par cellule sur le Z :
 - Min, max,
 - Nb de points dans la cellule
 - Valeur moyenne Z (meanZ)
 - Ecart type de la mesure
- Génération d'un fichier texte rassemblant les attributs par cellule
- Pour visualisation semis régulier de point à hauteur meanZ par cellule et génération d'un *.las pour la commodité de vision de contrôle, mais non nécessaire au calcul





2 : Carroyage, stats cellule



Seconde lecture en fichier, le traitement X, Y de chaque point modulo le pas de carroyage permet de classer le point dans la bonne tuile/cellule :

- Toute les tuiles ayant un ou plusieurs points du nuage sont élues
- Démographie sur chaque tuile :
 - Zmin, Zmax, nbre de points
 - Zmoy, σ_Z





3 : Filtrage Z et densité

- L'étape précédente fournit des stats (sur Z) par cellules
- Pas de sélection de points a priori, de nouveaux paramètres sont préparés
- On présume **benoîtement** d'une distribution normale (A approfondir)
- On retient 5 % de la population on fixe $Z_{\max} \leq (\text{MeanZ} - 2 \text{ sigma})$ c'est le sol
- On impose une densité minimale pour fiabiliser le calcul
- On travaille sur les cellules, pas d'accès disque pour ce filtrage
- Mise à jour de l'objet MAP mémoire C++ :
 - On retient les seules cellules à densité suffisantes,
 - On impose un $Z_{\max}$ comme indiqué $\Rightarrow Z_{\max} \leq (\text{MeanZ} - 2 \text{ sigma})$.
 - Cet objet MAP sert de référence pour une nouvelle lecture disque du fichier





4 : Relecture fichier avec de nouveaux critères

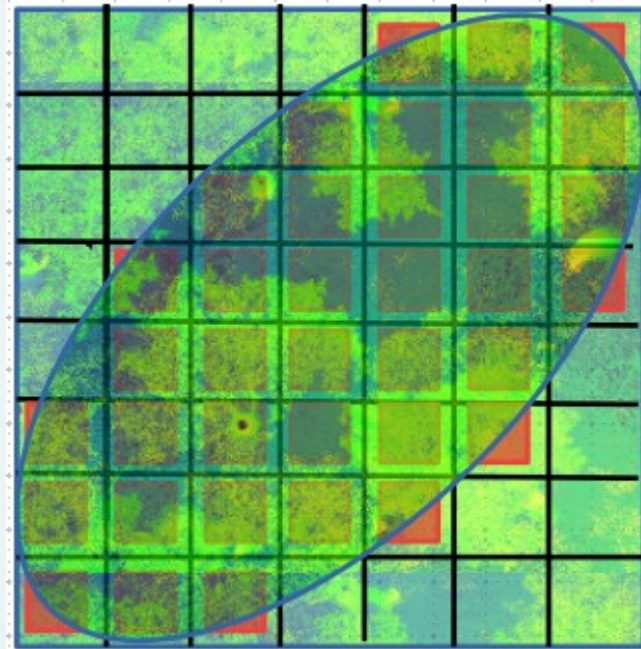
On procède maintenant à la **troisième lecture disque** du fichier

- Deux nouveaux critères ;
 - Le point doit se trouver dans une cellule éligible par densité
 - Sa coordonnées Z est $\leq$ au nouveau Zmax ($Z_{\text{mean}} - 2 \sigma_Z$)
- On repeuple ainsi une nouvelle carte (objet MAP C++) de cellule ne contenant que des points présumés appartenir au sol
- Les données statistiques Z_{mean} et σ_Z sont calculées à nouveau pour chaque cellule
- Nouvelle carte de cellule
- Quelques disparités avec cellules voisines adoucissement par filtre convolutionnel de type Sobel





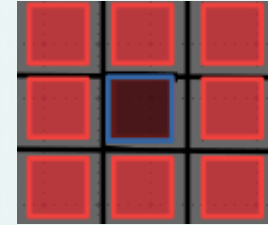
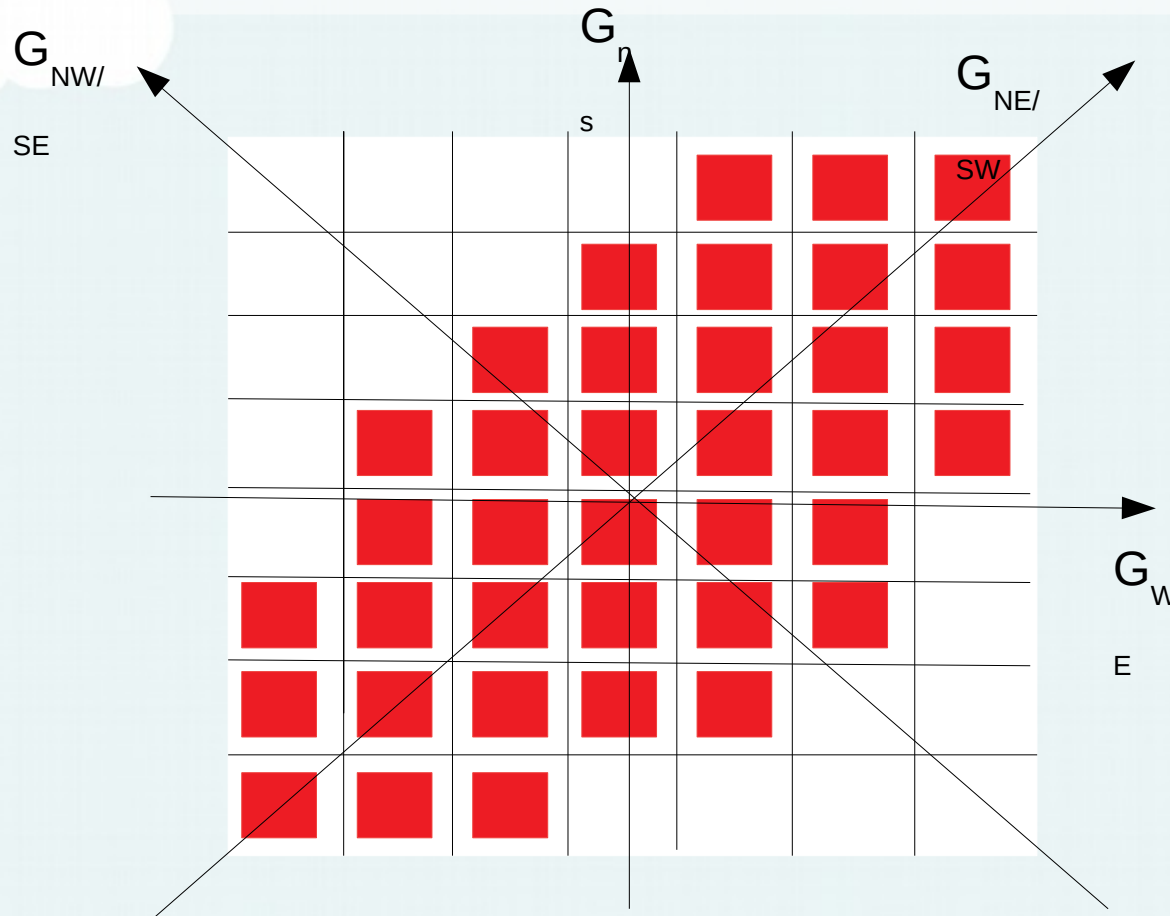
4 : Relecture fichier avec de nouveaux critères



On se retrouve avec des cellules suffisamment peuplées de points, dont les caractéristiques statistiques présument fortement de leur appartenance au sol.
Cependant des disparités de voisinage nécessite un filtre convolutif de premier ordre de type Sobel



Filtrage (3 x 3) avec gradient



Pour chaque cellule sélectionnée
La contribution retenue est
La sommes par couples opposés
si ils existent ces cellules voisines.

La cellule filtrée (noire) a une
Contribution pondérée au double
de celle du voisinage.

Le résultat final est normalisé



5 : Filtrage de lissage



- Filtre 9 cellules (3 x 3)
- On valide la cellule centrale en respect de la contribution pondérée des 8 voisins
- La pondération est obtenue par une évaluation grossière des gradients de dénivelé :
 - nord/sud, est/ouest
 - sud-est, nord-ouest
 - sud-ouest, nord-est
- On donne à la cellule calculée (centrale) le même poids contributif que la somme des contributions de chaque voisin
- On normalise en ramenant la contribution globale à 1



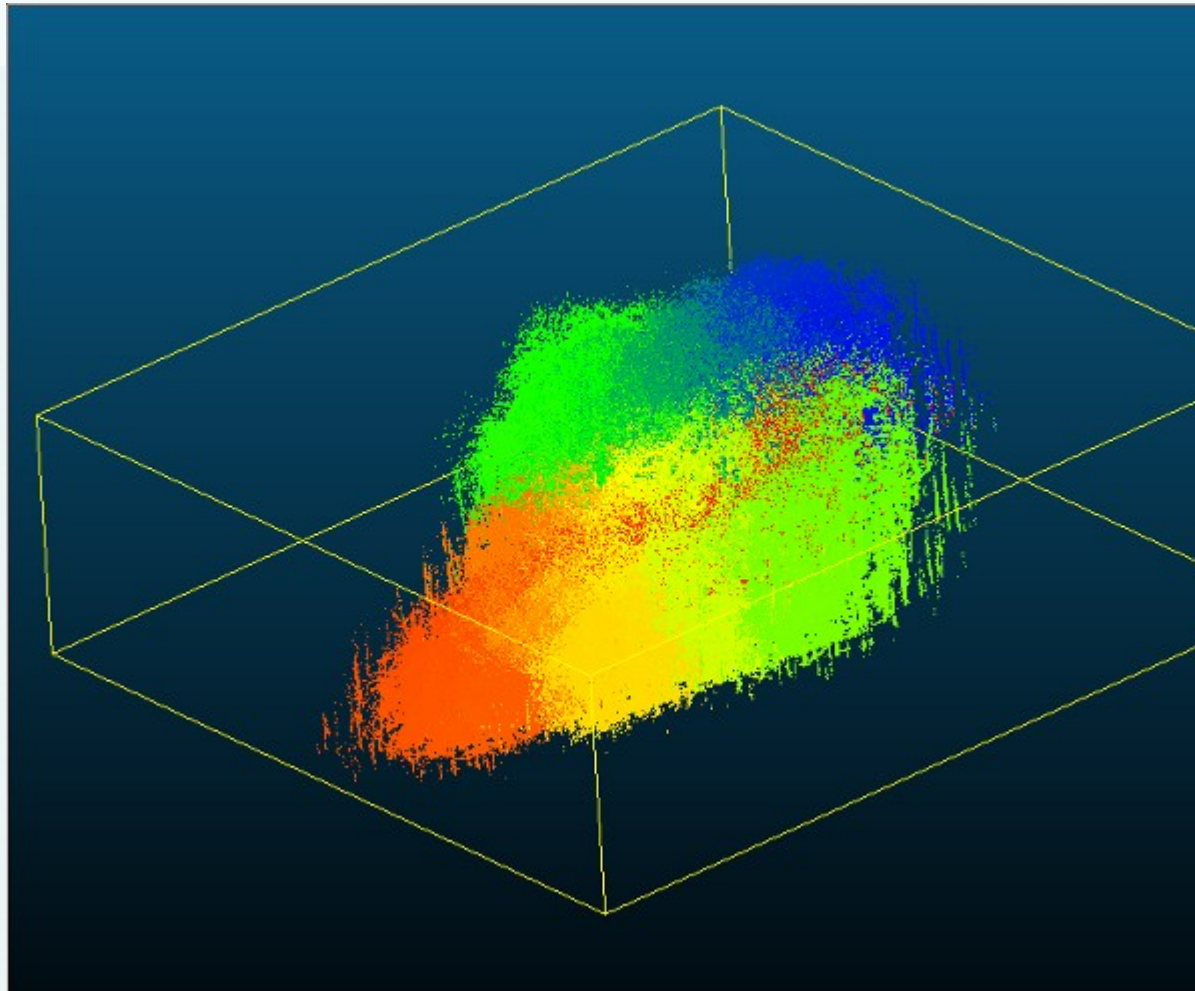
6 : Filtrage de voisinage KNN



- Filtre 9 cellules (3 x 3)
- On élit la cellule centrale si :
 - constat de mitoyenneté (plan xOy) avec au moins deux autres cellules,
 - différence d'altitude, entre cette cellule centrale et la cellule mitoyenne concernée, inférieure ou égale à la résolution au sol $\delta Z < R_{\text{és}}$.



VUE ISOMETRIQUE DE FORÊT (1/3)

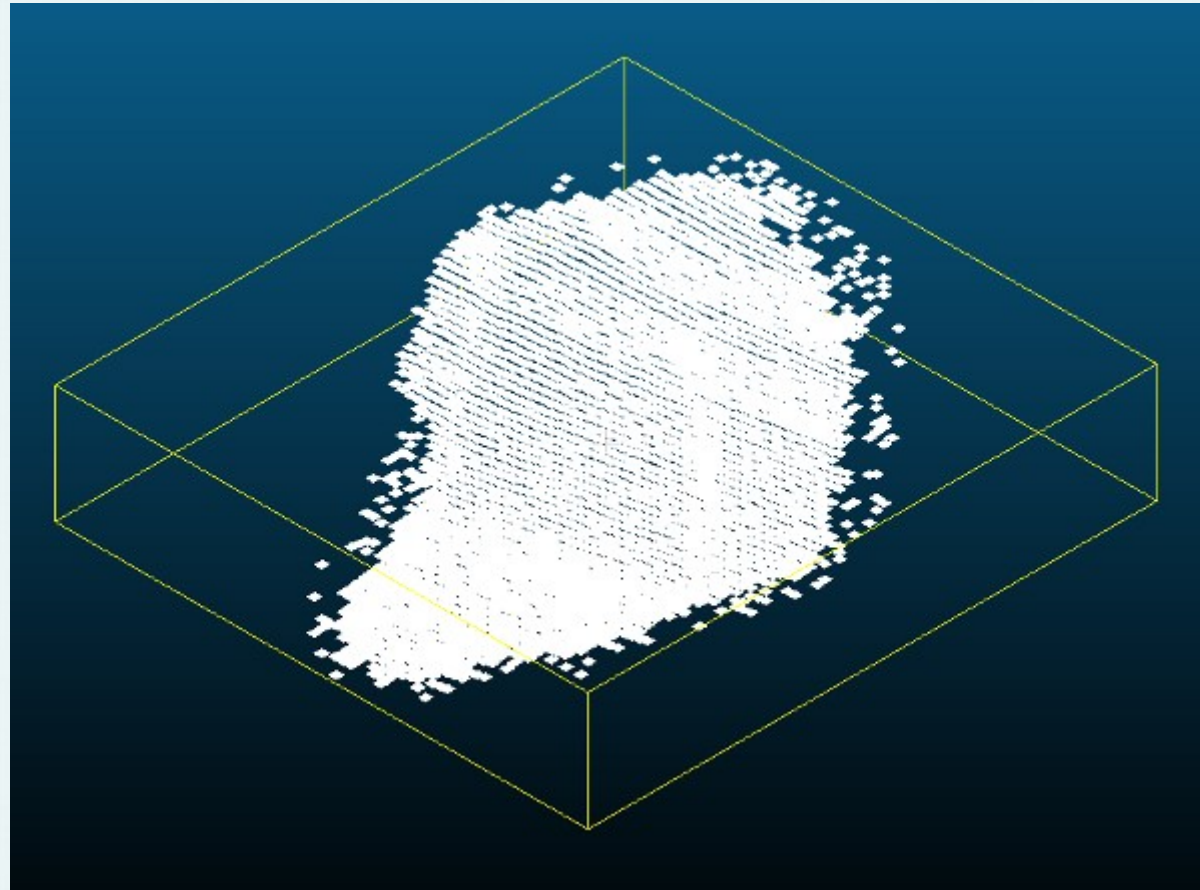


Vue isométrique d'une zone forestière de référence
Couleur significative du temps/heure de relevé.





VUE ISOMETRIQUE DE RELIEF (2/3)

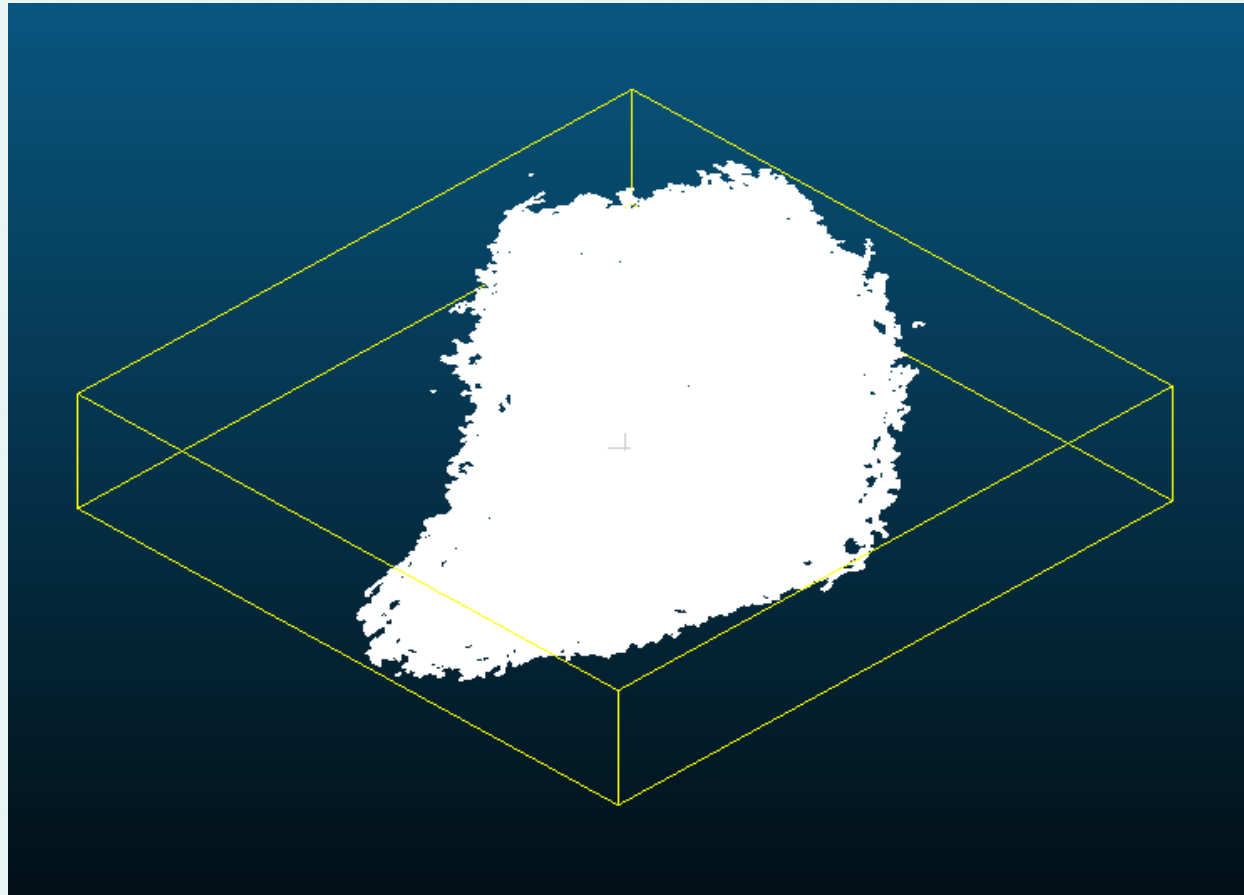


Vue du modèle de sol avec un dénivelé de 32 m en Z
Zone faiblement gauche en géologie (cf. écran 68)





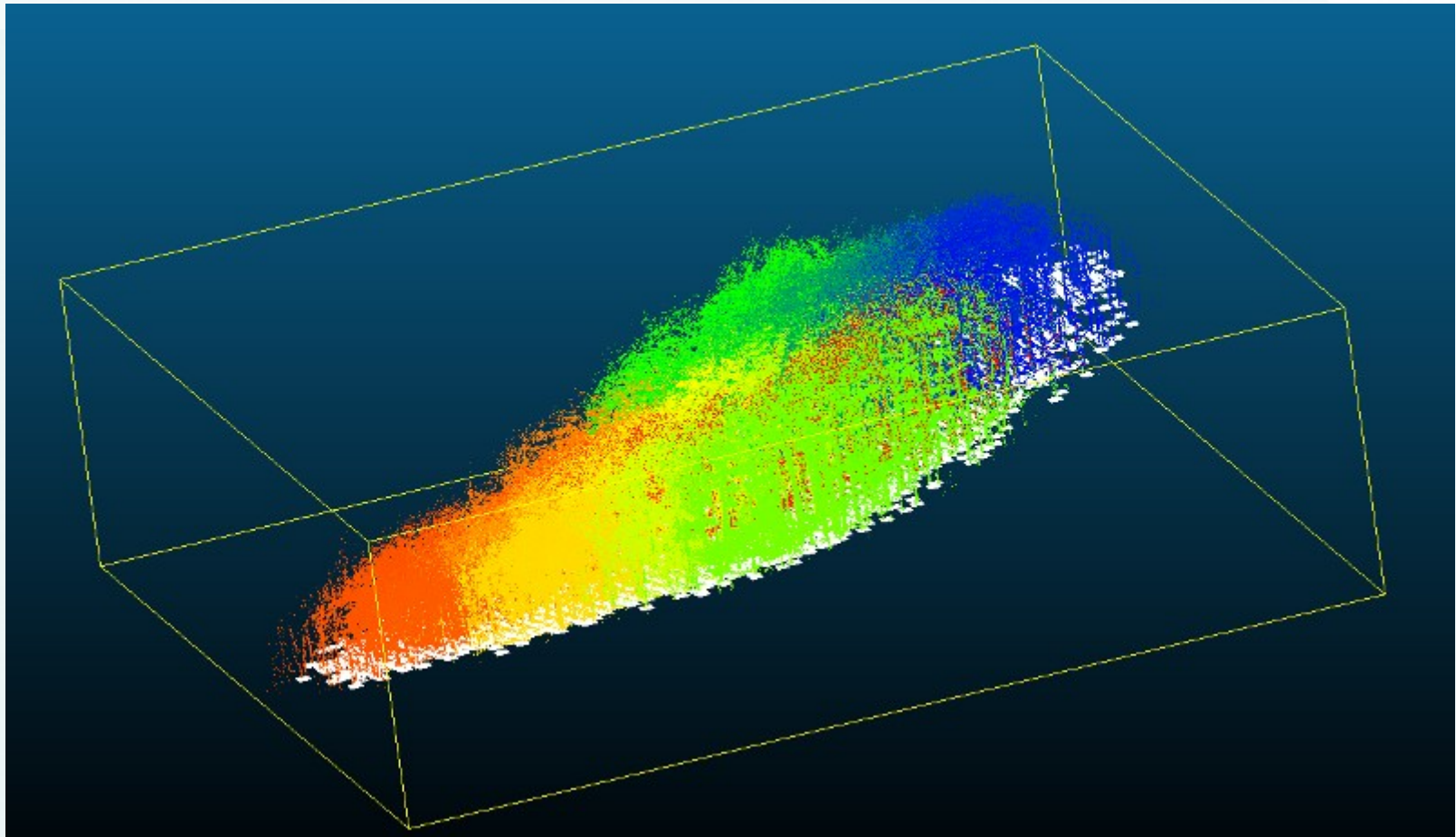
VUE ISOMETRIQUE DE RELIEF (3/3)



Modèle de sol, même zone, après application d'un filtre KNN : 2 cellules voisines à la courante et $\delta Z < R_{\text{rés}}$ pour éligibilité, ci-dessus cellules avec résolution de 50 cm.



VUE ISOMETRIQUE DE RELIEF + forêt

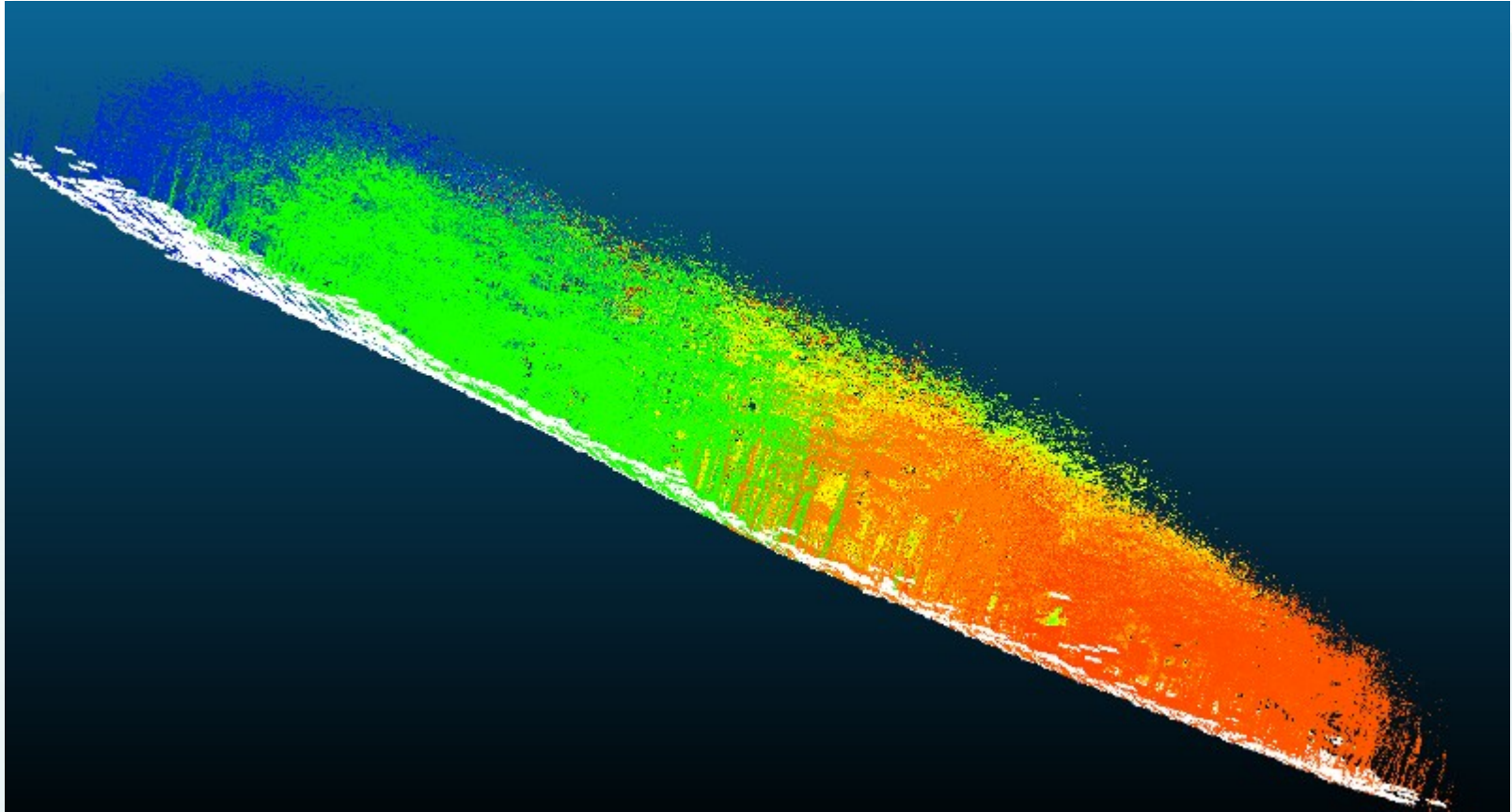


Habillage du modèle de terrain avec sa forêt, le dénivelé passe à 50 mètre ce qui est cohérent le modèle précédent





VUE ISOMETRIQUE DE RELIEF + forêt



Vue arrière, on voit la torsion gauche droite selon la diagonale principale qui gauchit la surface du terrain, (à gauche vue par en dessous du relief $z < 0$) à droite vue par au-dessus ($z > 0$)



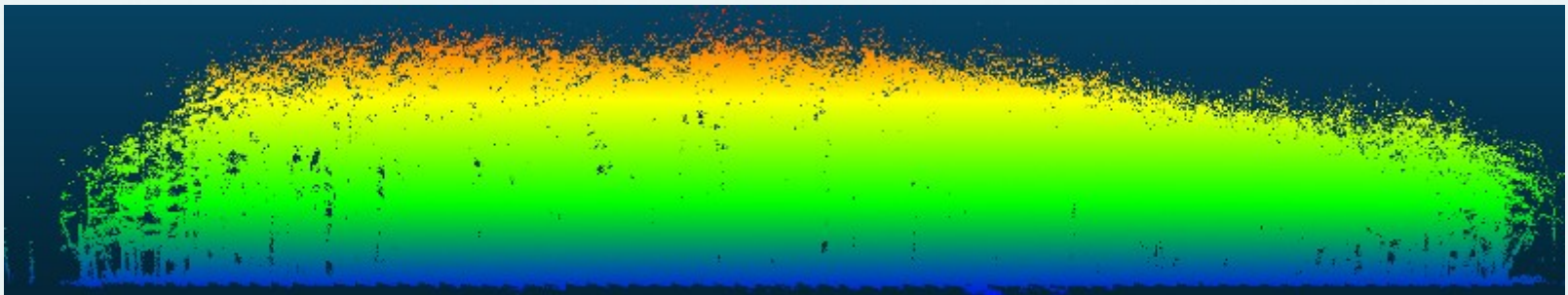
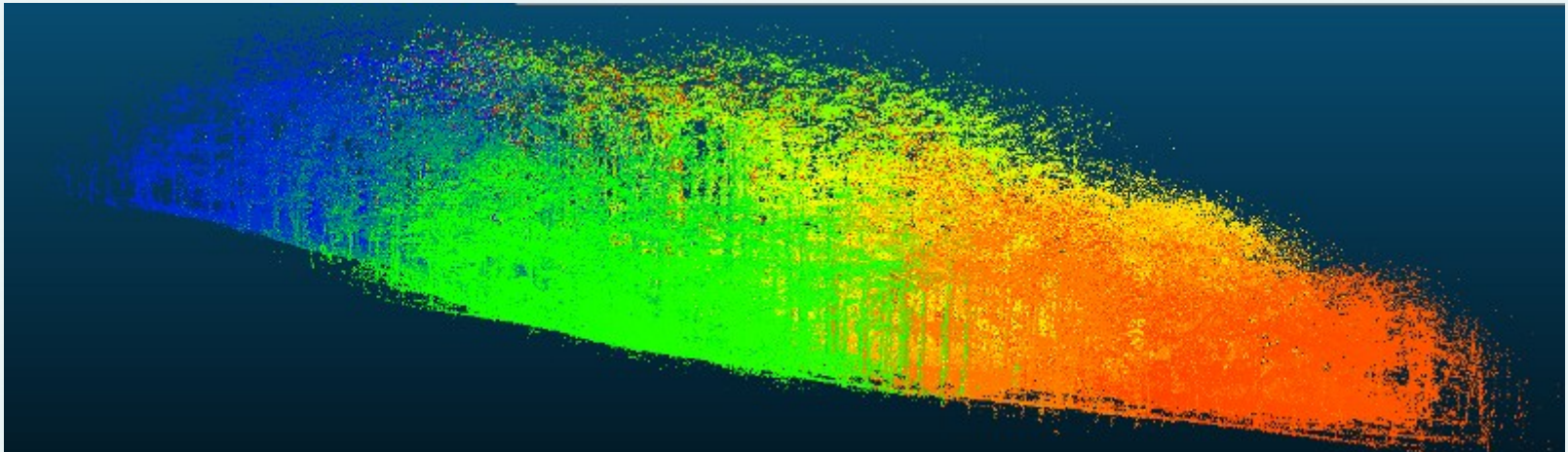
Mise au relief Zéro



- Par rapport au fichier initial nous disposons d'un modèle de terrain
- Ce modèle est validé sur certaines cellules
- Nous rejetons les autres cellules pour une qualité de mesures insuffisantes
- Le nouveau fichier est obtenu par soustraction du modèle de terrain aux données initiales,
- Il est validé sur les seules zones éligibles pour la poursuite du traitement, pour le reste la qualité d'acquisition est jugée insuffisante, l'acquisition de donnée est à reprendre
- La poursuite du traitement se fait par des découpes verticales paramétrées en Z
- Ceci nous fournit les sections dont nous avons à évaluer le positionnement X,Y et les autres attributs attendus.



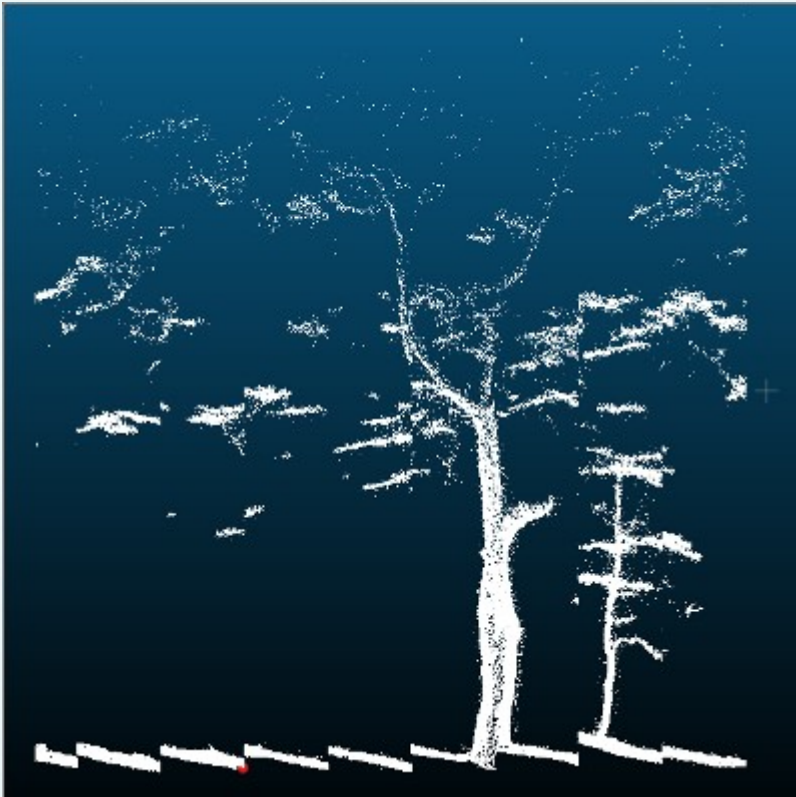
Mise au relief Zéro, vue arrière, avant et après retrait MNT





Exécution de coupes tomographiques

Maintenant que nous disposons d'un relief normalisé à Zéro, nous avons la possibilité de conduire des coupes tomographiques (sections), suivant les trois plans de coupe ZOY, ZOX, XOY en se déplaçant sur leur normale respective $\vec{I}_x, \vec{J}_y, \vec{K}_z$.



A gauche section verticale ($\vec{K}_x$) au-dessus section horizontale ($\vec{K}_z$), 30 cm d'épaisseur. On notera à gauche l'amélioration nécessaire du gradient de filtrage pour le MNT. A droite [60m x 35 m] les sections d'arbre => des petits pseudo cercles.

Ces sectionnements tomographiques nous permettent de construire des images *.png pour la conduite des algorithmes de classification et ou de reconnaissance de forme en section horizontale.



Modalité de localisation des arbres

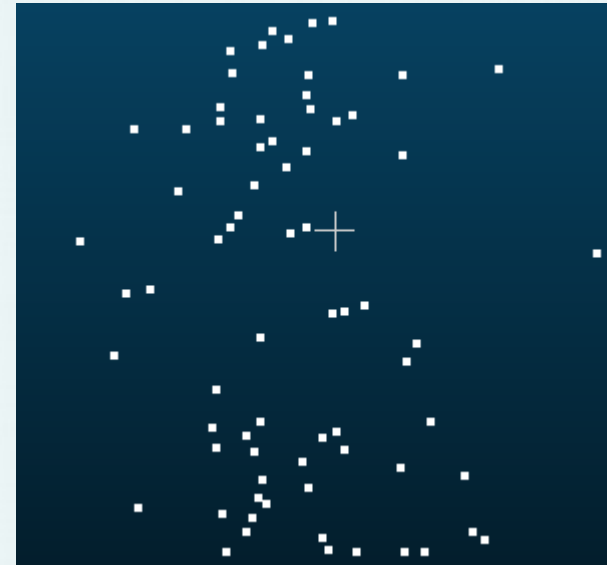
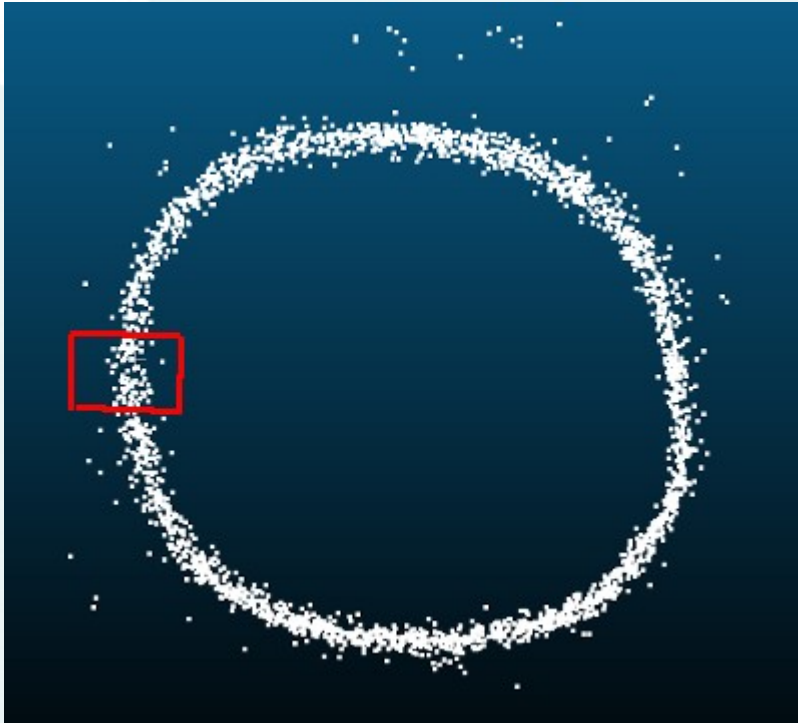


Les coupes tomographiques réalisées permettent de déboucher sur deux méthodes automatiques de localisation des arbres, que nous explorons :

- Les coupes verticales nous permettent de construire des histogrammes de densité dont les différents pics permettent de présumer de la présence d'un arbre donné dans le rectangle d'incertitude [largeur de la classe en X, épaisseur de la section de coupe en Y] ces statistiques sont directement extraites de l'analyse des fichiers tomographique *.las. Elles peuvent aussi présenter de l'intérêt pour localiser amers et homologues.
- Les coupes horizontales, mettent en évidence des sections d'arbre variables selon la hauteur, ces plans de coupe sont transformés en fichiers image (JPEG, PNG) sur lesquels nous appliquons une adaptation [**très forte pour ne pas dire abandon**] de la transformée de Hough circulaire, permettant d'identifier (localiser et mesurer) les pseudo cercles (enveloppe convexes) de sections d'arbre. Les fichiers images étant normalisés on retrouve aisément la localisation géographique précise.



Traitement des tomographies verticales identification des Sections d'arbres, géolocalisation, attributs géométriques



Une section type d'arbre $\sim \phi = 1 \text{ m}$, zone marquée $\sim 81 \text{ cm}^2 \Rightarrow 75$ points projetés au sol $\Rightarrow 15$ pts pour 16 cm^2 , tentative de création d'image PNG avec une résolution de 4 cm, soit une image de 1500 px X 875 px pour la section horizontale précédente



D'un nuage laser (3D) à une image PNG (2D)



Les trois écrans qui suivent reprennent au calage près et dans l'ordre:

- Un nuage de point en vue de dessus d'une épaisseur en Z de 30 cm, il s'agit d'une copie d'écran,
- Le même nuage de point projeté dans le plan XOY (image PNG extraite par calcul de projection au sol du nuage de points pour une section de 30 cm à une hauteur donnée), les objets d'intérêt sont les sections pseudo circulaires qui correspondent aux troncs d'arbres sectionnés verticalement, le reste étant des échos de feuilles ou branches voire du bruit de mesure,
- Le même nuage de point nettoyé et présentant le seuls objets d'intérêt dont on extrait automatiquement, la position (X,Y,Z) transformable en RFG 93, et le diamètre selon le filtre employé.



Sectionnement vertical sur relief zéro (1)



Ecran CloudCompare vue de dessus d'une section LIDAR horizontale de 60 m x 35 m à $Z = 1.5$ m avec épaisseur 0.3 m sur zone boisée (pied altitude 0)



Sectionnement horizontal sur relief zéro (2)



Projection en Image PNG res 2cm de la vue préc. une section horizontale de 60 m x 35 m à Z =1.5 m avec épaisseur 0.3 m sur zone boisée (pied altitude 0)



Sectionnement horizontal sur relief zéro (3)

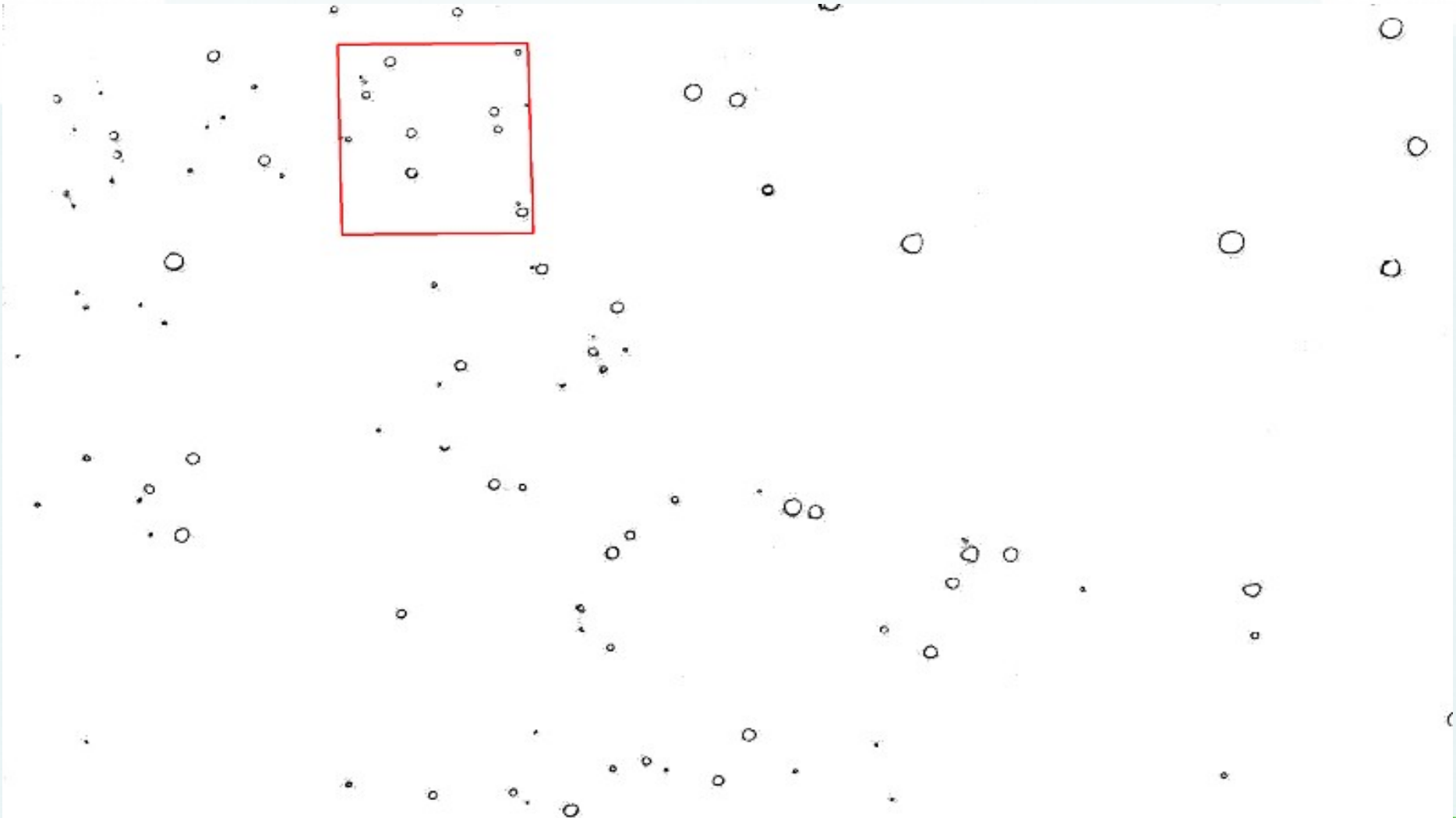
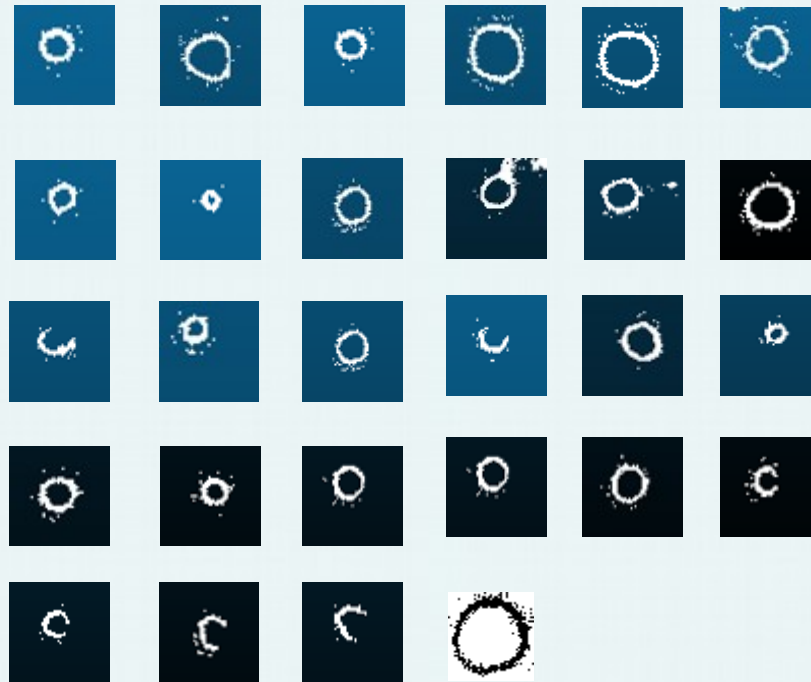


Image débruitée, normalisée, avec la zone de contrôle de test (voir écran 81/82)



1ère bibliothèque de formes de sections

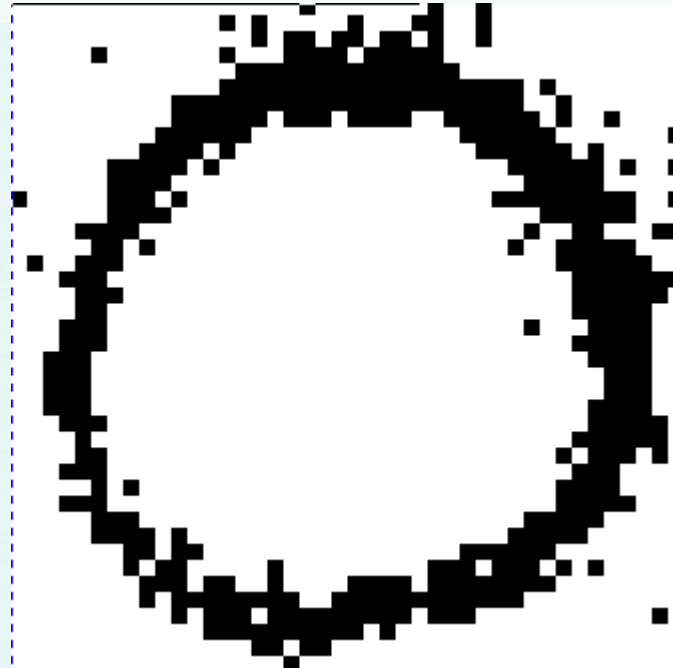


Cette première bibliothèque (type 1 qualitative/Forme) est construite par prélèvement immédiat dans les copies d'écran, sous CC. Une seconde (Type 2) sous forme d'images calibrées en échelle de gris (quoique binaire 0/255) à haute résolution (Ex. 44px X 44 px ~ 250 pts lidar 23cm x 23 cm), plus exploitables et quantifiables car c'est une projection des points LIDAR sur le plan Z=0. Type 3 faite par classification automatique (à venir et à apprécier).

Perspective immédiate => localisation par corrélation, bibliothèque de type 2



Traitement en mode 2D par corrélation directe



Les section d'arbres projetées au sol sont des section pseudo circulaires dont les collections d'enveloppes convexes permettent de construire des bibliothèques de filtres afin d'extraire la localisation des arbres et donnant automatiquement la fourchette de diamètre. Une couronne circulaire optimisée au moindre carrée donne aussi de bons résultats

Ci-dessus un exemple **de filtre** une Matrice de corrélation 42 x 42px



Caractéristique du filtre



Les constructions d'images sont totalement normalisées :

- Elles sont calibrées dimensionnellement.
- Dans le cas présent elles sont construites avec une résolution de 4 cm, ce qui explique l'incertitude de mesures de ± 5 cm.
- Par ailleurs chaque point est géoréférencé (mais se reporter à la remarque du dernier écran)
- Le filtre a deux caractéristiques qui lui sont attachées :
 - La fourchette de détection de diamètre à ± 4 cm
 - Le nombre de pixels qu'il contient, il est construit au pixel près, bien qu'il soit en échelle de gris, il ne prend que les valeurs 0 et 255, donc il est binaire. Ce nombre de pixel permet de normaliser l'image résultat (en fait un histogramme local) qui est une image raster, de manière à obtenir un pic d'intensité de détection avec un voisinage similaire pour des diamètres identiques.



Mode de fonctionnement du filtre

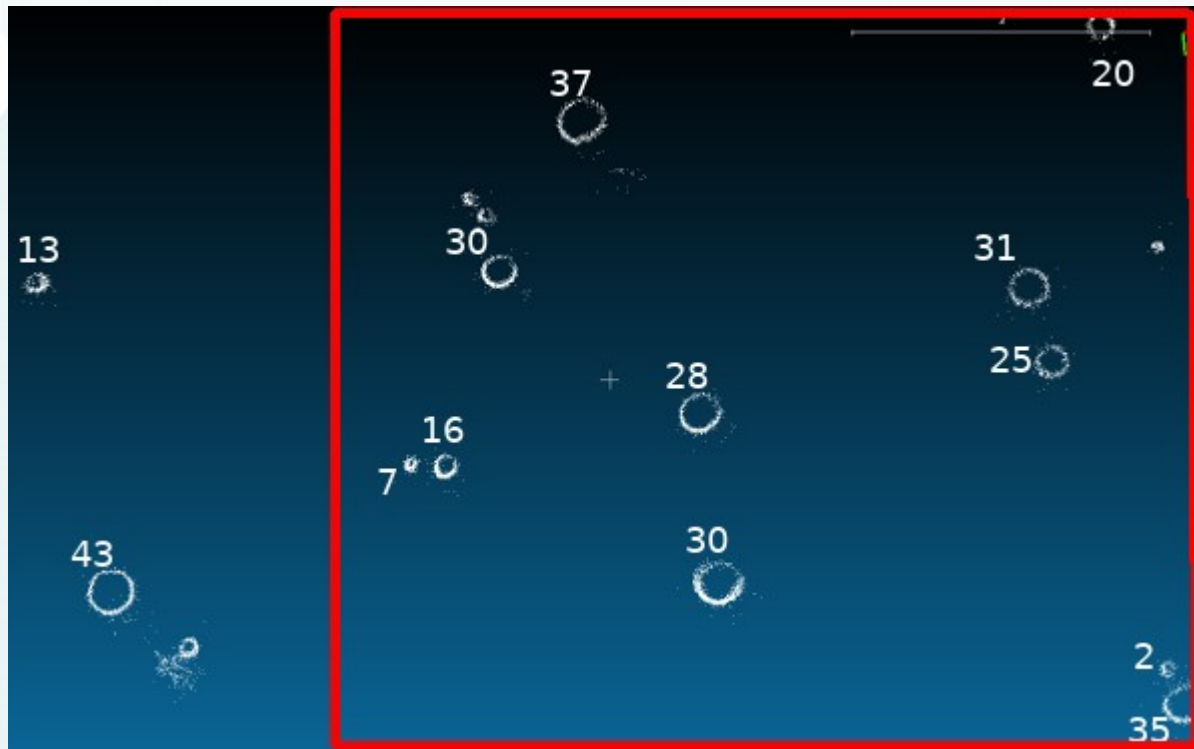


- C'est un filtre glissant
- Le principe de filtrage repose sur du comptage de pixels communs entre le filtre et l'image à une position de pixel donnée de l'image et non sur la contribution pondérée du voisinage, **c'est de la détection binaire et du comptage aboutissant à un histogramme surfacique**, il peut être présenté comme une image raster dont les pics soulignent une présomption de présence d'arbre.
- En sortie nous sélectionnons un nombre paramétrable de maximum qui donnent la localisation de l'arbre, le filtre étant caractéristique de son diamètre, à la hauteur de la section considérée
- Comme dans la transformée circulaire de Hough un vote / un seuil permet de retenir les arbres éligibles à une localisation donnée





Vue de dessus d'une section de nuage lidar

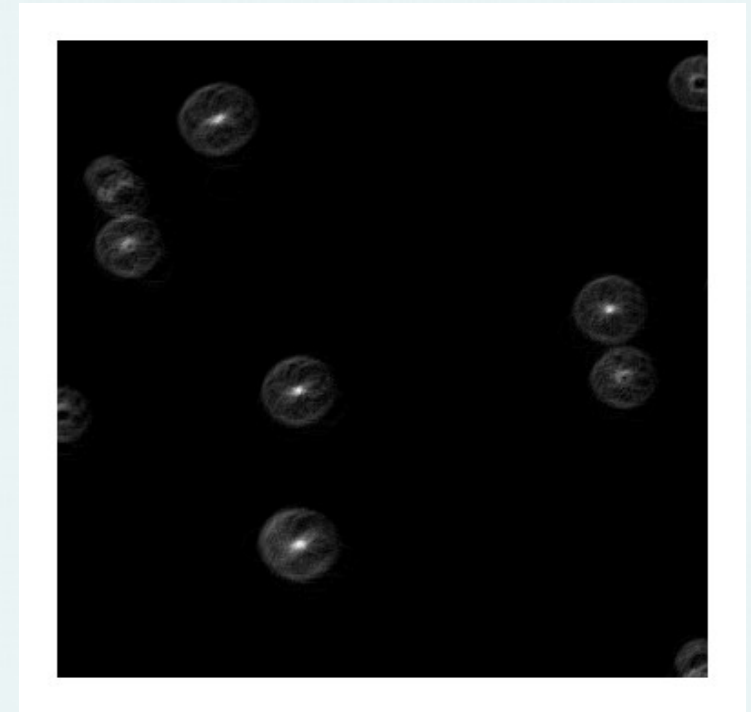
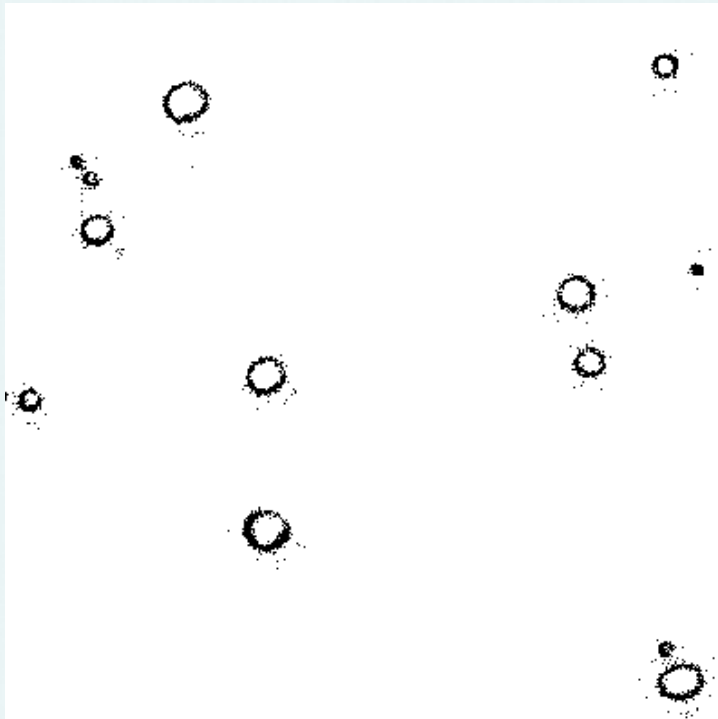


Détail du nuage LIDAR

Les section d'arbres projetées au sol sont des section pseudo circulaires dont les collections d'enveloppes convexes permettent de construire des bibliothèques de filtres afin d'extraire la localisation des arbres, ci-dessus vue de dessus d'une section horizontale de nuage de points. Les chiffres près des enveloppes convexes donnent le diamètre évalué en cm. L'encadrement rouge délimite la zone de résultats donnés ci après. Ci-dessus copie d'écran CloudCompare.



Résultat de filtrage

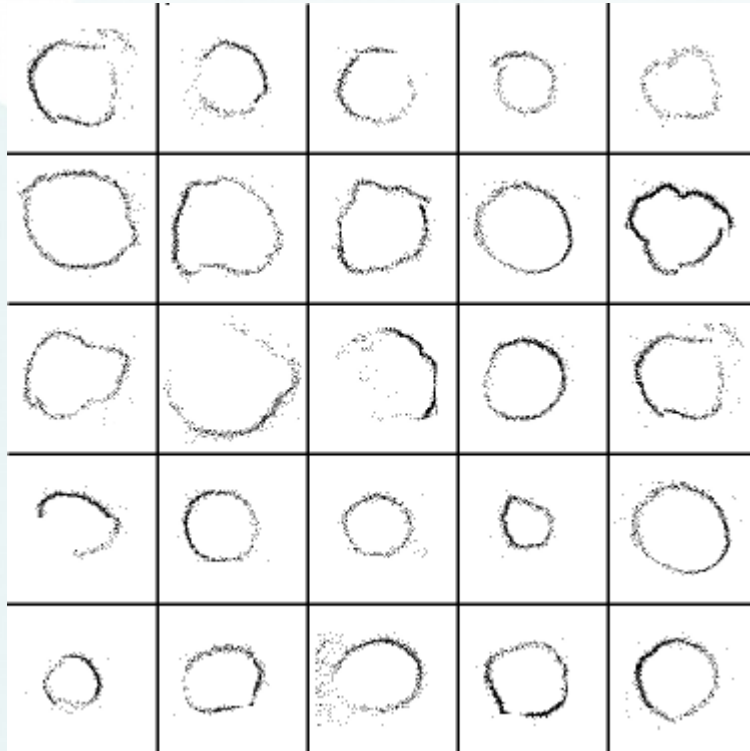


A gauche l'image *.png calculée de la section du nuage de point (cf. cadre rouge précédent) à droite le résultat après filtrage de corrélation. A droite l'image résultat après filtrage pour une filtre 32 cm (± 5 cm). Les images étant géométriquement normalisée la localisation est fournie dans le résultat.

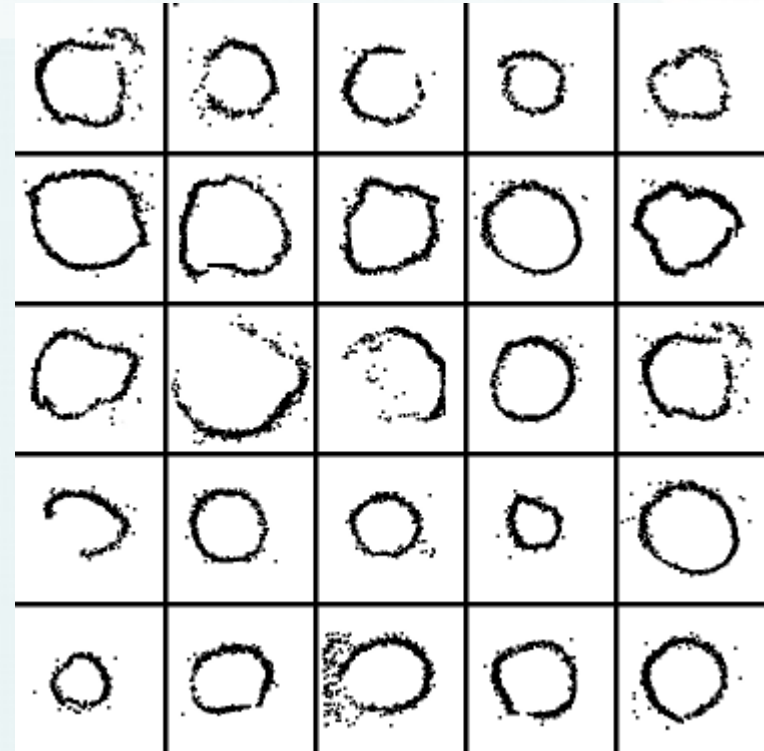




Traitement Préalable sur l'image : densité



Forme initiale de section PNG



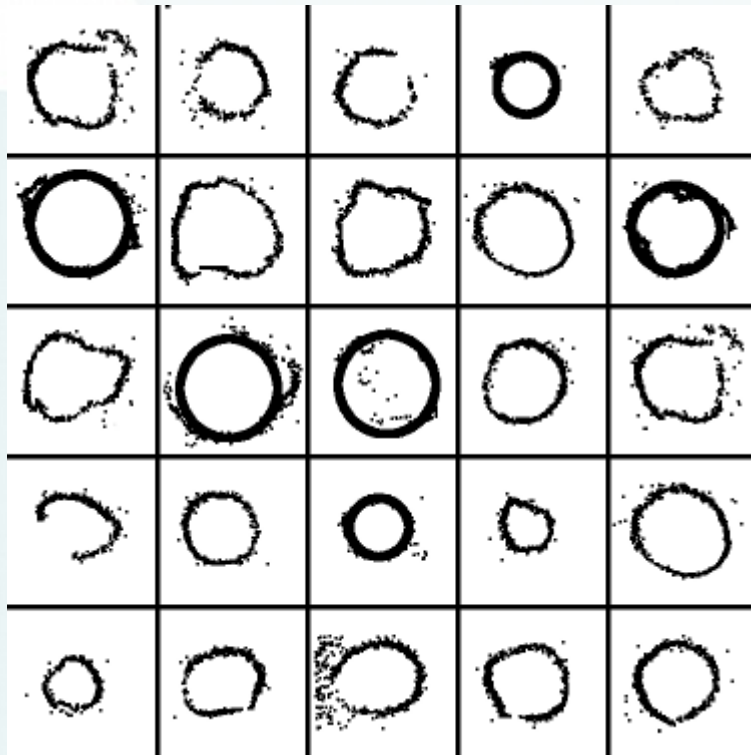
Normalisation de densité par érosion

Toutes les sections d'arbre ne sont pas circulaires, les formes sont variées. Selon la qualité du relevé de mesure, on rencontre deux autres problèmes, la faible densité de Points collectés et la non fermeture des enveloppes convexes de section. Un filtre Morphologique d'érosion normalise la densité.



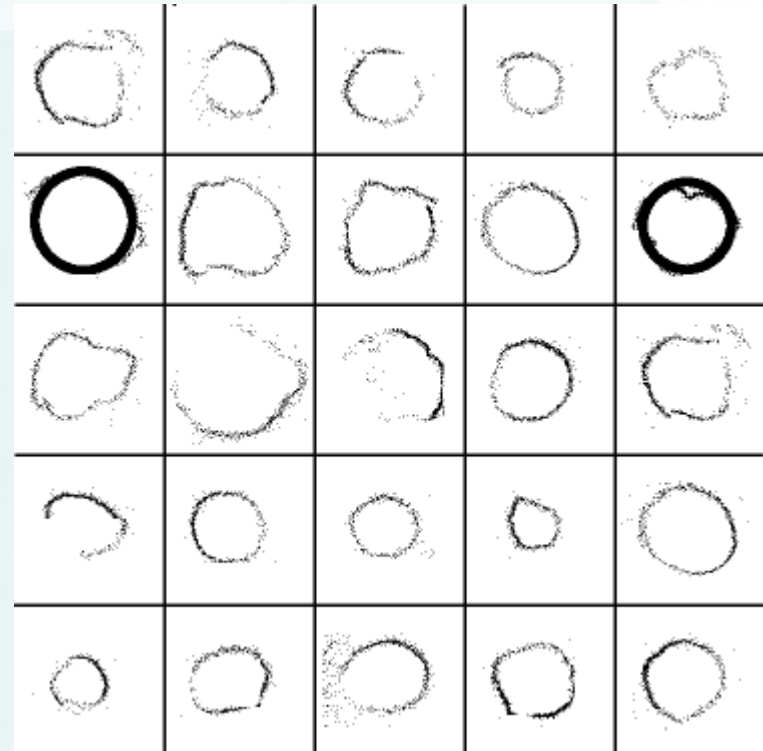


Traitement Préalable sur l'image : fermeture



Filtrage sur densité normalisée

Φ 100 cm, 80 cm, 60 cm



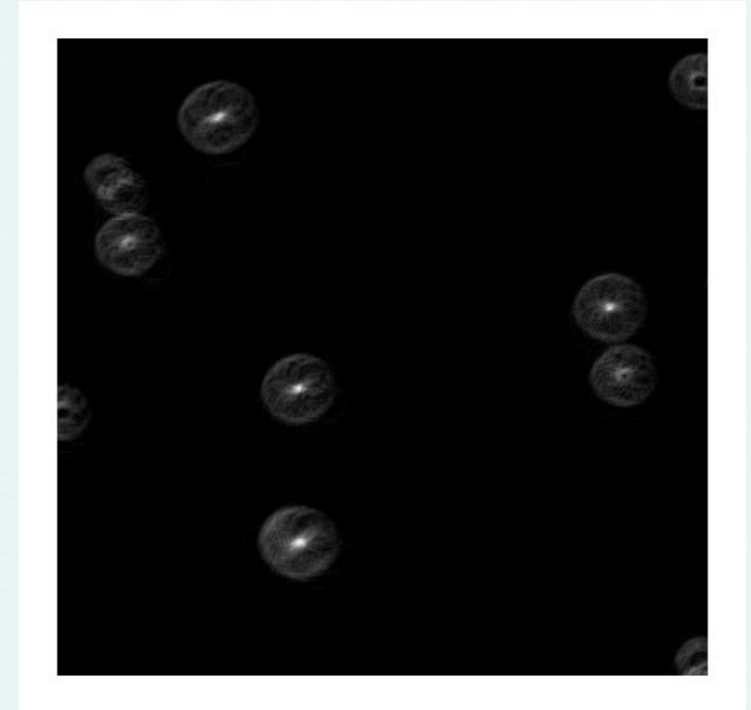
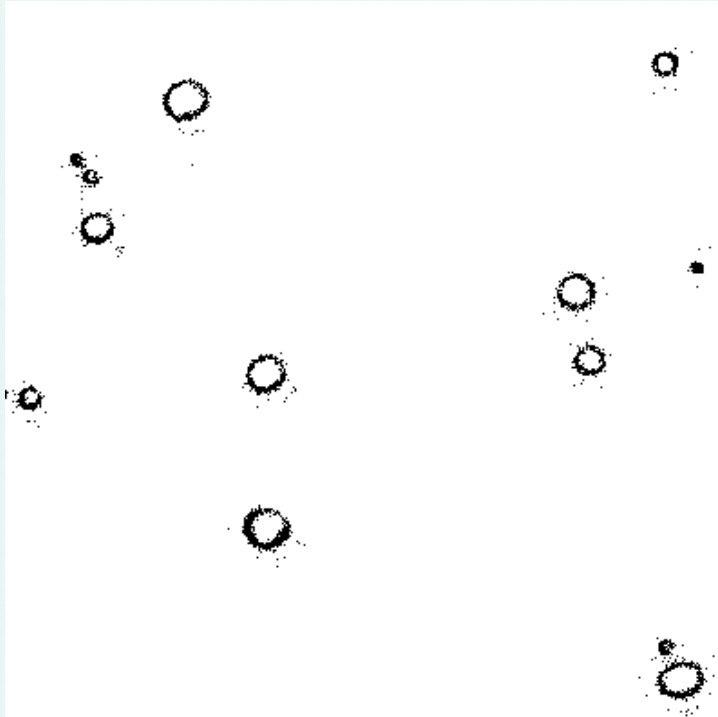
Filtrage sur données initiales

Φ 100 cm, 90 cm

Un filtre circulaire de convolution compense la non fermeture si la densité est suffisante



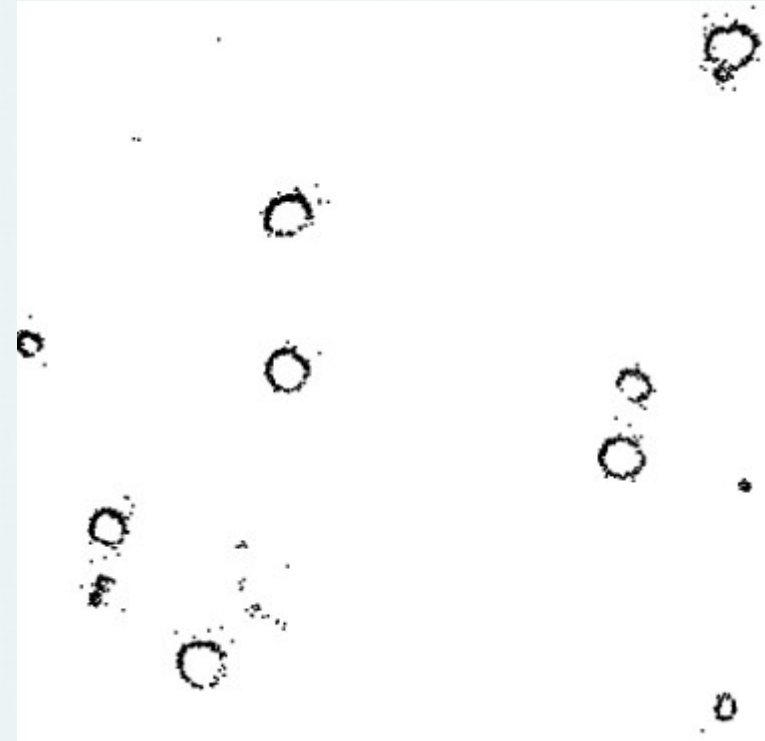
Résultat de filtrage



A gauche l'image *.png calculée de la section du nuage de point (cf. cadre rouge précédent) à droite le résultat après filtrage de corrélation. A droite l'image résultat après filtrage pour une filtre 32 cm (± 5 cm). Les images étant géométriquement normalisée la localisation est fournie dans le résultat.



Zone d'intérêt avant et après érosion



L'opération morphologique d'érosion permet d'accroître la densité de point ce qui permet d'accroître l'effet discriminant du filtrage de convolution



Modalité du filtrage : accélération



- En prévision d'application future l'image est enregistrée en niveau de gris [0, 255]
- Son exploitation se fait en mode binaire, 0 ou 255
- La corrélation se limite à du comptage pas de pondération
- Détection des zones vides sous le masque de filtrage => saut équivalent à la taille du filtre
- Détection des zones à faible densité => saut équivalent à la 1/2 taille du filtre
- Résultat obtenu => histogramme surfacique sur les zones retenues pour l'analyse
- Résultat image raster de contrôle (usage qualitatif, intérêt limité)

Le centre estimé de l'arbre est un agrégat de points, l'appartenance à cet agrégat Est le fruit d'un vote sur seuillage du nombre de points rencontrés dans la zone.

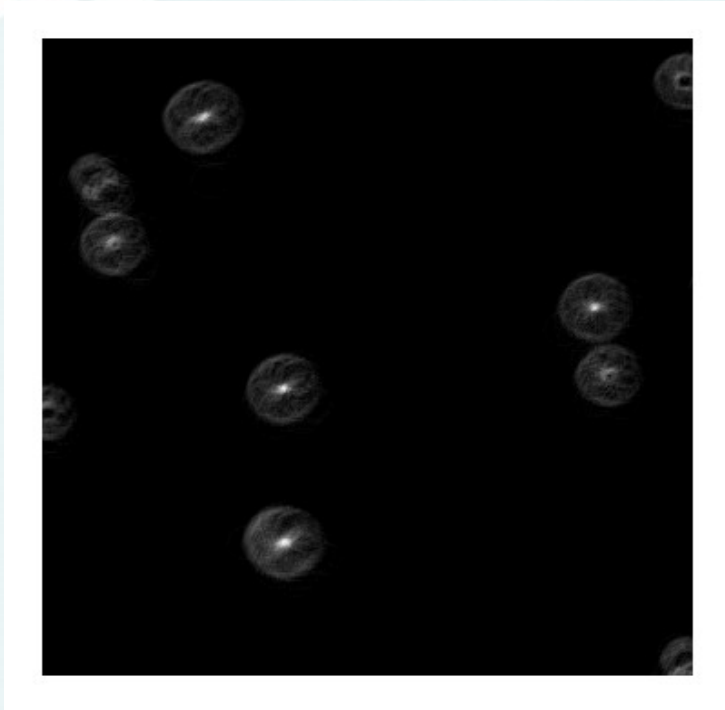
Le résultat est une liste [x, y, Nbre de points] qui peut être présentée comme une image raster limitée au voisinage des maximum, ce qui en fait une image réduite de la corrélation globale.

Un traitement de voisinage effectué sur cette liste donne la position exacte de chaque arbre





Corrélation : globale VS accélérée



Les temps de calculs sont réduits d'un facteur 100 avec threads, à évaluer avec C_CUDA. Complexité totalement linéaire liée au nombre de pixels de l'image.

A gauche image raster de contrôle balayage global

A droite image raster de contrôle balayage des seules zones à densité suffisante

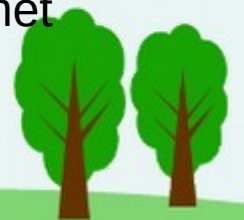




Rappel relevés amers lors des mesures (1/5)

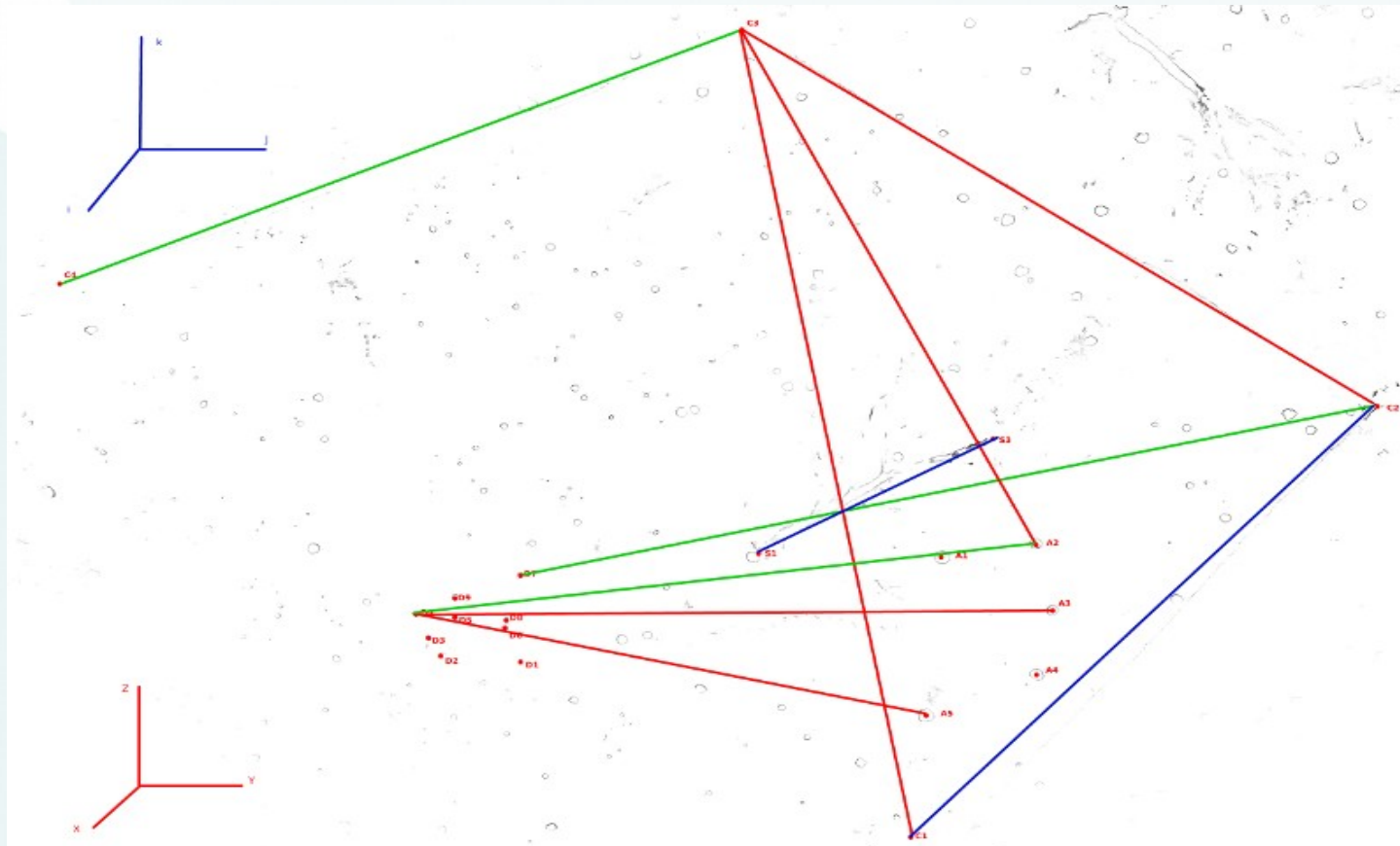


Ici nous présentons notre zones expérimentales de mesures. A des fins de contrôle nous avons relevé la localisation de nombreux points (21) ce qui nous permet de disposer de 20! vecteurs alors que 2 nous suffisent. Il s'agit de valider la fiabilité de calcul de la matrice de transformation.





Rappel relevés amers lors des mesures (2/5)



La construction de vecteurs possibles sur la zone, à partir des amers. La précision des calcul sera d'autant meilleure que la taille relative des vecteurs par rapport à la zone est importante et leur angle d'intersection (entre vecteur inter-amer et homologue) proche de 90° . Pour que les calculs soient valides, il faut respecter la règle du trièdre direct, tant pour les produits scalaires, vectoriels ou quaternions.



Rappel relevés amers lors des mesures (3/5)



En complément des écrans 49 et 50, et suite aux deux écrans précédents construits après un premier dépouillement, rappelons que :

- Les relevés d'amers sont à faire durant la campagne de mesures,
- Les localiser dans la zone de mesure où une réception GPS est possible
- Il faut s'assurer d'une bonne densité de points sur ces amers, pour pouvoir identifier aisément leurs homologues dans le nuage final,
- A priori 3 points suffisent (2 vecteurs) un 4ème (6 vecteurs possibles) permettra de valider la qualité et la précision du relevé,
- Il faut s'assurer de la plus grande distance possible entre points dans la zone et éviter toute colinéarité.





Rappel relevés amers lors des mesures (4/5)

Retrait des segments aberrants, constatés par comparaison entre les données du nuage retenus comme amers et les relevés sur site

- Les calculs de positionnement géométrique relatif des segments homologues deux à deux vont permettre de calculer la matrice de transformation (rotation et translation),
- La fiabilité de cette matrice résultante est liée à la qualité des mesures conduites,
- Nous retenons comme premier postulat qu'à cette phase du traitement le nuage de points est considéré comme fiable dont peu déformé,
- Nous retenons comme second postulat que tout segment aberrant est le fruit d'une anomalie de mesure GPS sur site,
- Un segment est déclaré aberrant si il y a une différence de longueur $> 1 \%$ (**marge d'erreur**) entre le segment relevé sur **site** et son homologue dans le **nuage de points**, alors le couple sera rejeté pour toute conduite de calcul de la matrice de transformation,
- La transformation exige de faire coïncider un couple de l'espace **site** avec un couple de l'espace **nuage de points**, un second contrôle est l'identité des angles formés par les couples retenus dans leur espace d'origine à la marge d'erreur près.
- N points relevés sur site permettent de construire (N-1)! Segments, (N-2)! couples.
- Dans le cas présent 1 seule mesure sur site, 20 pts GPS, sans précautions fortes, donne (1 % 5 couples), (2 % 9 couples), (3 % 13 couples), (4 % 19 couples), (5 % 20 couples).
- Le contrôle de l'angle entre couple de vecteurs, fait apparaître des anomalies en mesures.
- Il faut deux couples pour une matrice de transformation, selon la marge d'erreur tolérée un moyennage des matrices est envisageable (24 pour 1 % d'erreur).





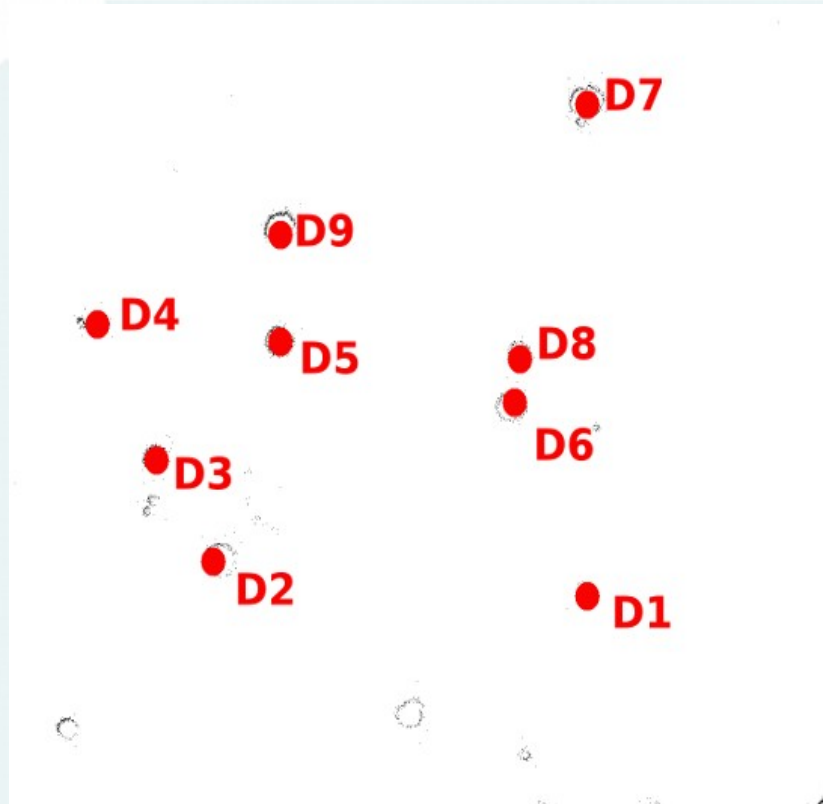
Rappel relevés amers lors des mesures (5/5)

ANCRAGE DES ROTATIONS DE CALAGE SUR UN POINT UNIQUE TROUVER LE MEILLEUR POINT !

- Cet aspect est fondamental pour l'évaluation de la matrice de rotation.
- La validation des relevés sur le terrain par rapport à leurs homologues dans le nuage de points se fait selon deux critères, après construction des segments :
 - À l'erreur tolérée près, l'identité de longueur des segments,
 - À l'erreur tolérée près, l'identité de leur angle d'intersection.
- Comme le recalage se fait sur la base de deux rotations rigides, il est nécessaire de trouver le point d'ancrage, qui doit être unique, de cette rotation :
 - La tolérance étant respectée, tant en longueur qu'en angle.
 - Trouver les couples de segments ayant un point commun, avec le maximum de longueur et le plus grand angle possible d'intersection.
 - Ce point commun sélectionné constituera l'ancrage de la rotation et sera également la référence de translation.
 - Sur ces bases calculer la matrice globale des deux rotations.



On retrouve les amers initiaux



On retrouve les amers initiaux, ainsi que l'ensemble des autres arbres extraits et classés selon le diamètre de section après le filtrage de corrélation. On notera également l'effet de miroir à propos duquel il faut être précautionneux lors de la construction de l'image PNG. De l'image PNG il faut respecter les règles de parcours d'image de libPNG.





Liste de voisinage et Extraction des positions

{171.0, 325.0, 142}
 {217.0, 153.0, 143}
 {217.0, 154.0, 155}
 {217.0, 155.0, 142}
 {218.0, 154.0, 147}
 {302.0, 154.0, 146}
 {302.0, 155.0, 143}
 {303.0, 153.0, 155}
 {303.0, 154.0, 165}
 {303.0, 155.0, 156}
 {303.0, 156.0, 149}
 {304.0, 153.0, 158}
 {304.0, 154.0, 165}
 {304.0, 155.0, 163}
 {304.0, 156.0, 149}
 {305.0, 153.0, 141}
 {305.0, 154.0, 145}
 {305.0, 155.0, 144}

La liste donne chaque pixel (X, Y, nbPoints) corrélé avec le filtre et le nombre de points trouvés au voisinage.

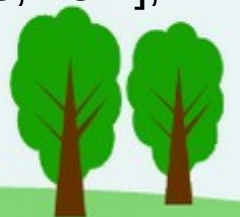
Reconstruction des groupements / cumuls de voisinages de ces points (distance, à un point pivot, inférieure $< d$) d étant un estimatif de la distance inter-arbre ($\geq 120\%$ du rayon du filtre).

La moyenne de chaque groupement/cumul identifié positionne la section d'un arbre de diamètre donné par le filtre à ± 5 cm.

Cumul X $\Rightarrow$ 171, Cumul Y $\Rightarrow$ 325, Nb points $\Rightarrow$ 142,
 Nombre de cumuls $\Rightarrow$ 1 Position arbre [X, Y] $\Rightarrow$ [171, 325],

Cumul X $\Rightarrow$ 869, Cumul Y $\Rightarrow$ 616, Nb points $\Rightarrow$ 587,
 Nombre de cumuls $\Rightarrow$ 4 Position arbre [X, Y] $\Rightarrow$ [217, 154],

Cumul X $\Rightarrow$ 3947, Cumul Y $\Rightarrow$ 2007, Nb points $\Rightarrow$ 1979,
 Nombre de cumuls $\Rightarrow$ 13 Position arbre [X, Y] $\Rightarrow$ [303, 154],





Fin du traitement et mise au référentiel RFG93

PROCESSUS DE PREPARATION et de MISE EN OEUVRE DES MATRICES DE TRANSFORMATION (s → source, t → target(cible), T → Translation)

- Nous travaillons sur un nuage source dont nous calons des points sur un nuage cible, les deux nuages sont dans un même référentiel,
- Les amers de terrains dans le nuage cible, leurs homologues dans le nuage source
- Sélection des couples de vecteurs optimum ($\vec{V}_1, \vec{V}_2$) avec un point commun (ancrage de transformation) soit trois points, un couple des vecteurs d'amers dans le nuage cible ($\vec{V}_{1t}, \vec{V}_{2t}$) et un couple de vecteurs homologues dans le nuage source ($\vec{V}_{1s}, \vec{V}_{2s}$) .
- Translater l'ancrage cible sur l'ancrage source $\vec{T}_{ts}$ et le tout sur le zéro $\{0, 0, 0\}$ $\vec{T}_{s0}$ du référentiel qui devient l'ancrage des rotations (origine des vecteurs sources et cibles), l'axe de la 1ère rotation est la normale $\vec{N} = \vec{V}_{1t} \otimes \vec{V}_{1s}$.
- Calculer ainsi la matrice de transformation superposant $\vec{V}_{1s}$ avec $\vec{V}_{1t} \Rightarrow [M1]$,
- Transformer l'extrémité de $\vec{V}_{2s} \rightarrow \vec{V}_{2sT}$, $\vec{V}_{2sT} = [M1] * \vec{V}_{2s}$, calculer la matrice de transformation superposant $\vec{V}_{2sT}$ avec $\vec{V}_{2t} \Rightarrow [M2]$, l'axe de rotation est $\vec{V}_{1t}$ (+ calcul 2 vecteurs ortho à $\vec{V}_{1t}$). l'angle de rotation entre les lignes projections ortho des extrémités $\vec{V}_{2t}$ et $\vec{V}_{2sT}$ sur $\vec{V}_{1t}$
- La matrice globale de transformation $[M2 * M1] = M$
- l'opération finale de transformation des points source vers les points cible est définie par l'opération suivante :

$$[PointCible] = [M] * [PointSource - \vec{T}_{s0}] - \vec{T}_{ts}$$



Fin du traitement et mise au référentiel RFG93



Les opérateurs de transformation étant définis, nous arrivons à minimiser la quantité de calculs car nous appliquons ces opérateurs **aux seuls points de localisation des arbres** dans la zone (quelques centaines) contre l'ensemble des points du nuage (quelques dizaines de millions par ha)

Comme cela est indiqué les images sont normalisées (position pixel linéairement liée à la géolocalisation RFG93) :

- Un point de position retrouve sa position [X,Y] dans le nuage de points (référentiel lidar).
- La position Z du terrain, à la position, est retrouvée dans la cellule correspondante du MNT.
- La matrice de transformation recalcule par rotation l'ensemble des points de localisation dans le référentiel RFG93.
- Enfin la translation est effectuée en calcul mode entier afin de limiter les erreurs, vu le nombre de chiffres significatifs.
- Les données de l'arbre, ses attributs en final, sont donnés dans un format permettant la mise en base de données PostGis/Postgres ou SpatiaLite.



Performances pressenties et industrialisation



A l'exception :

- du calcul des paramètre de la matrice de rotation qui est écrit en Python,
- des rotations écrites en C_CUDA,

tout est développé en C++ sous forme de module de commande en ligne. Certains filtrage seront ré-écrits sous C_CUDA et la parallélisation possible des tâches sera complétée.

Les traitements les plus lourds sont :

- La génération du MNT,
- Le filtrage des coupes tomographiques horizontales avec l'extraction des positions.

En l'état actuel des développements, le processus complet aboutissant à l'extraction des positions d'arbres pour chaque filtre de diamètre est ~ **2 minutes par ha.**

L'industrialisation, intégrera ces traitements et proposera un enchaînement des tâches de manière plus conviviale, s'appuiera sur PyQGIS pour l'interface homme/machine.



Performances pressenties et industrialisation



L'essentiel des applicatifs est développé en C++ sous forme de modules de commande en ligne. Certains filtrage sont écrits en C_CUDA et la parallélisation possible des tâches sera complétée. Les traitements les plus lourds sont :

- La génération du MNT,
- Le filtrage des coupes tomographiques horizontales avec l'extraction des positions.

En l'état actuel des développements, le processus complet aboutissant à l'extraction des positions d'arbres pour l'ensemble des filtres de diamètre demande ~ **20 minutes par ha**. Le processus partant du fichier brut ***.las** issu du relevé, jusqu'à la génération de la couche attributaire QGIS des arbres avec pour chacun (identité, géoloc. RFG93, diamètre section à une hauteur donnée), et la requête SQL de mise en base de données géométrique de ces informations.

L'industrialisation, intégrera ces traitements et proposera un enchaînement des tâches de manière plus conviviale et s'appuiera sur PyQGIS pour l'interface homme/machine.



Outils complémentaires



- PrJextractTile extraction bornée de volume de points .
- CreateMNT_optimal Création de MNT avec accélérateur.
- FiltreCircleStd Filtre de comptage accéléré et extraction des positions d'arbre.
- NeighBourHoodCluster Recherche de voisinage
- Png-X-Y-zmoy projection d'une section LIDAR en image PNG
- SubtractMNT Addition / soustraction MNT
- MapNoOrderPrj reconstruction MNT
- Extract_Tile_from_las extraction de parallépipède de points
- Create PNG-X-Y-Zmoy création PNG depuis un sectionnement horizontal de nuage de points
- AdaptPng binarisation d'image PNG 0/255





Conclusions et poursuite des travaux

- En matière d'algorithmique et de traitement **toutes les étapes sont parcourues**, depuis le nuage brut de départ jusqu'à l'extraction de l'arbre géolocalisé avec ces attributs pour mise en base de données SIG.
- La méthode de l'année dernière (Travaux Yasser Boutelab & Marcel Bariou) donnant aussi des résultats, mais incomplète (faute de temps) sera reprise notamment pour l'apprentissage automatique qui n'est pas abordé ici, concernant la détection et la classification des formes au sol comme des cavités et parasites sur arbre (champignons).
- Tous les modules élémentaires spécialisés sont ré-écrits en C++ , voire en C_CUDA
- La robustesse de l'ensemble est à vérifier avec des retours terrains
- L'intégration globale est à faire avec PyQGIS qui peut aisément appeler tous ces modules avec un interface plus convivial, ce qui est encore beaucoup de travail, a priori sans grandes surprises.

Mais, mais mais ... des des amers géolocalisés dont les points homologues sont parfaitement identifiables dans des nuages de points restent toujours nécessaires et désirés.

En matière techno. Les travaux d'intégration et de parallélisation (Cuda et OpenMp) des exécutables, restent de très bons sujets pour des alternants ou des stagiaires ingénieurs ou M2, sans oublier des algos d'accélération à enrichir.



Récapitulatif : le pipeline du processus



1 - Acquisition sur terrain, données LIDAR et amers géolocalisés (lots de mesures de 15 à 20 mn). [Lidar Geoslam +GPS]

2 - Construction du fichier *.las et validation, taille : 1 à 3 Go conditionnée par la durée de la mesure. [Logiciel Geoslam + CloudCompare]

3 - Extraction du modèle numérique de terrain (pas de 50 cm) sur lot. Après transformation éventuelle (11) elle constituera l'empreinte au sol en S.I.G. [CreateMNT_KNN_Optimal_CM]

4 - Mise des pieds d'arbre des lots au relief Zéro (orthoprojection par retrait M.N.T). [Substract_MNT_CM]

5 - Extraction d'une tranche par lot, de 10 cm d'épaisseur, à 1.30 m de hauteur, parallépipède, de points lidar, parallèle au sol. [Prj_Extract_Tile]

6 - A partir des tranches extraites des lots création de fichiers images (*.png) par orthoprojection sur XOY. Résolution de construction 1cm / 1 px. [Create_PNG-X-Y-Zmoy]

7 - Nettoyage des fichiers PNG et fermeture (75%) des enveloppes convexes de sections d'arbre : opération **manuelle**, faiblesse algo, à approfondir. [GIMP]

8 - Erosion morphologique, normalisant la densité, sur l'image *.png. [GIMP, G'MIC]

9 - Application des batteries de filtres décroissants en taille (120 cm à 20 cm), détection des pivots d'accumulation, effacement des sections détectées. [FILT_CIRCLE_STANDARD]

10 - Élimination des doublons (pivots accumulation), avec arbitrage de diamètre sélectionné, extraction Z arbre dans MNT, attributs par section. [FILT_CIRCLE_STANDARD]

11 - Calcul des matrices de transformation, à partir des amers sélectionnés sur chaque lot. [WG84_2_RFG93]

12 - Transformation des géolocalisations (WGS84 => RFG93) des sections détectées par la mise en œuvre des matrices précédentes pour chaque zone et préparation des données attributaires pour mise en S.I.G. [Forest_Data_For_SIG]

