



# Détection automatique et semi-automatique de fissuration sur le platier rocheux normand à partir de données LiDAR

*Mehdi El Mouatadil, Cyrille Fauchard*

*26 septembre 2019*

*Insa Rouen/LETG-Geophen/ROL/Cerema/M2C*



## **Contexte**

*Projet **Topochronic** – étude diachronique, trait de côte et fissuration*  
*Stage Insa Génie Mathématiques 3eme année réalisé par **Mehdi El-Mouatadil***

## **Objectif du projet**

*Comment traiter de grand volumes de données sur le trait de côte? Peut-on en extraire des indices pertinents comme la fissuration pour comprendre et prévoir les phénomènes d'érosion.*

## **Objectif particulier du stage**

*A partir de données LiDAR sur Villerville, Fécamp et Dieppe, **cartographier la fissuration** du platier rocheux.*

## *Objet de l'étude*

*Qu'est-ce que la fissuration?*



*Saint-Marguerite-sur-Mer*

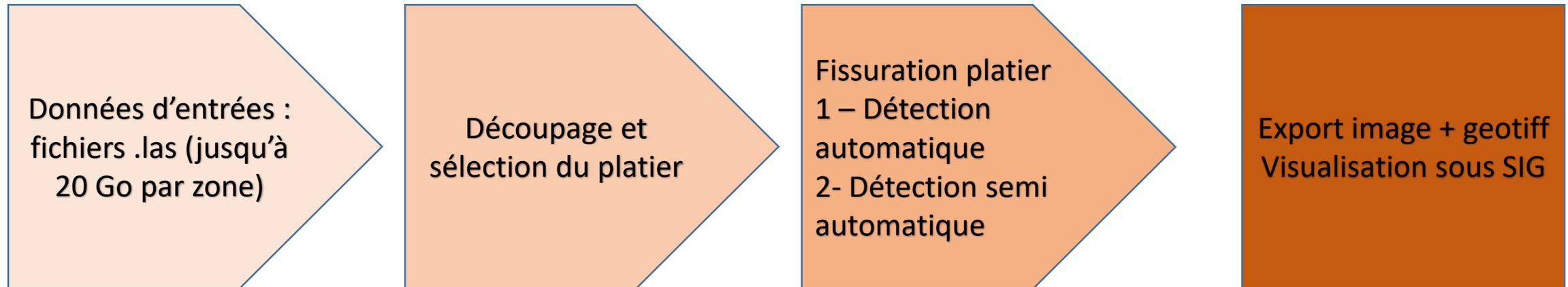


*Saint-Marguerite-sur-Mer*



*Viller-sur-Mer*

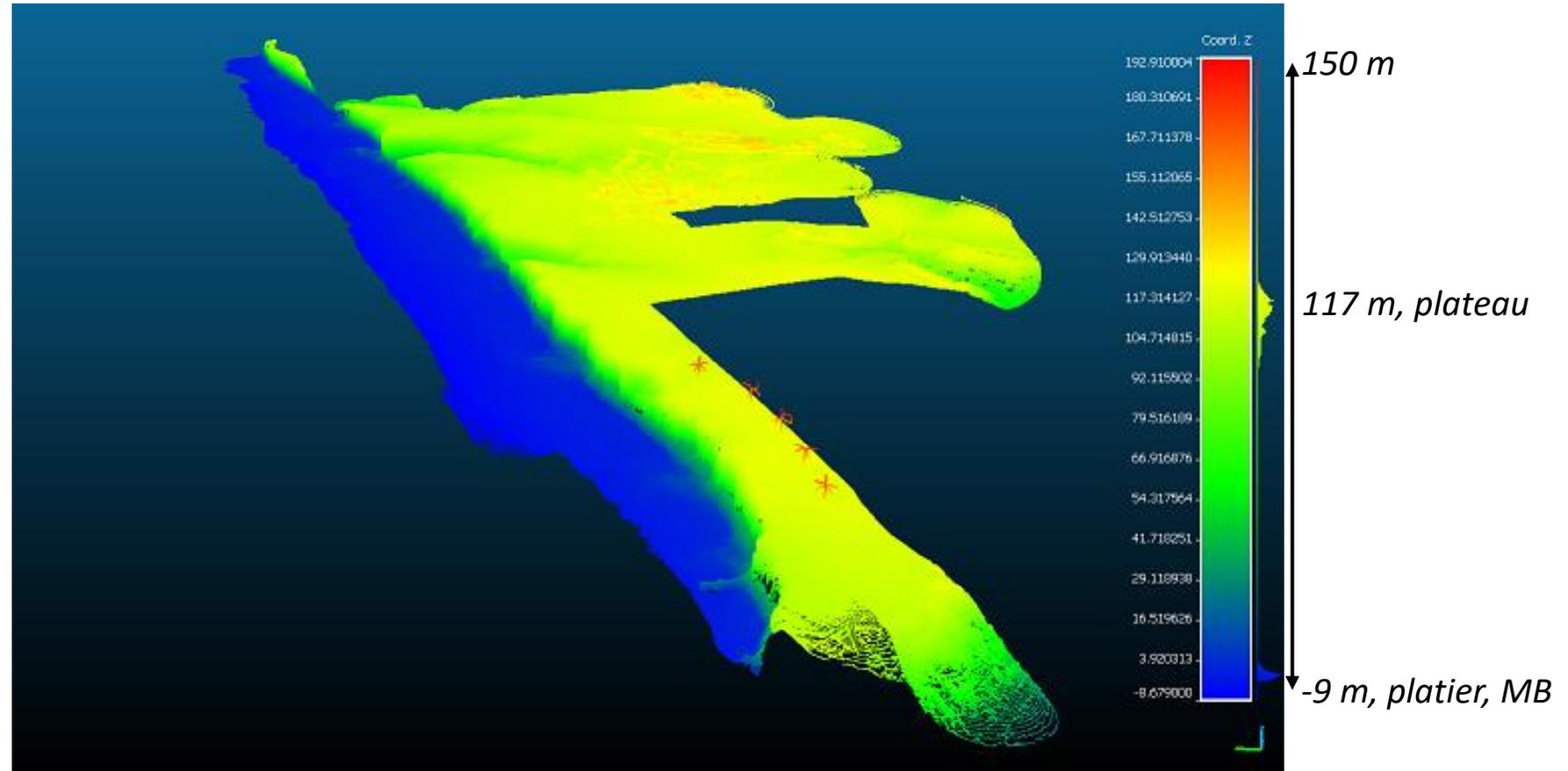
## ***Méthodologie adoptée***



## Données ROL

Données d'entrées  
: fichiers .las  
(jusqu'à 20 Go par  
zone)

Lecture directe des  
fichiers .las sous  
**Python** puis  
découpage. Affichage  
sous **CloudCompare**

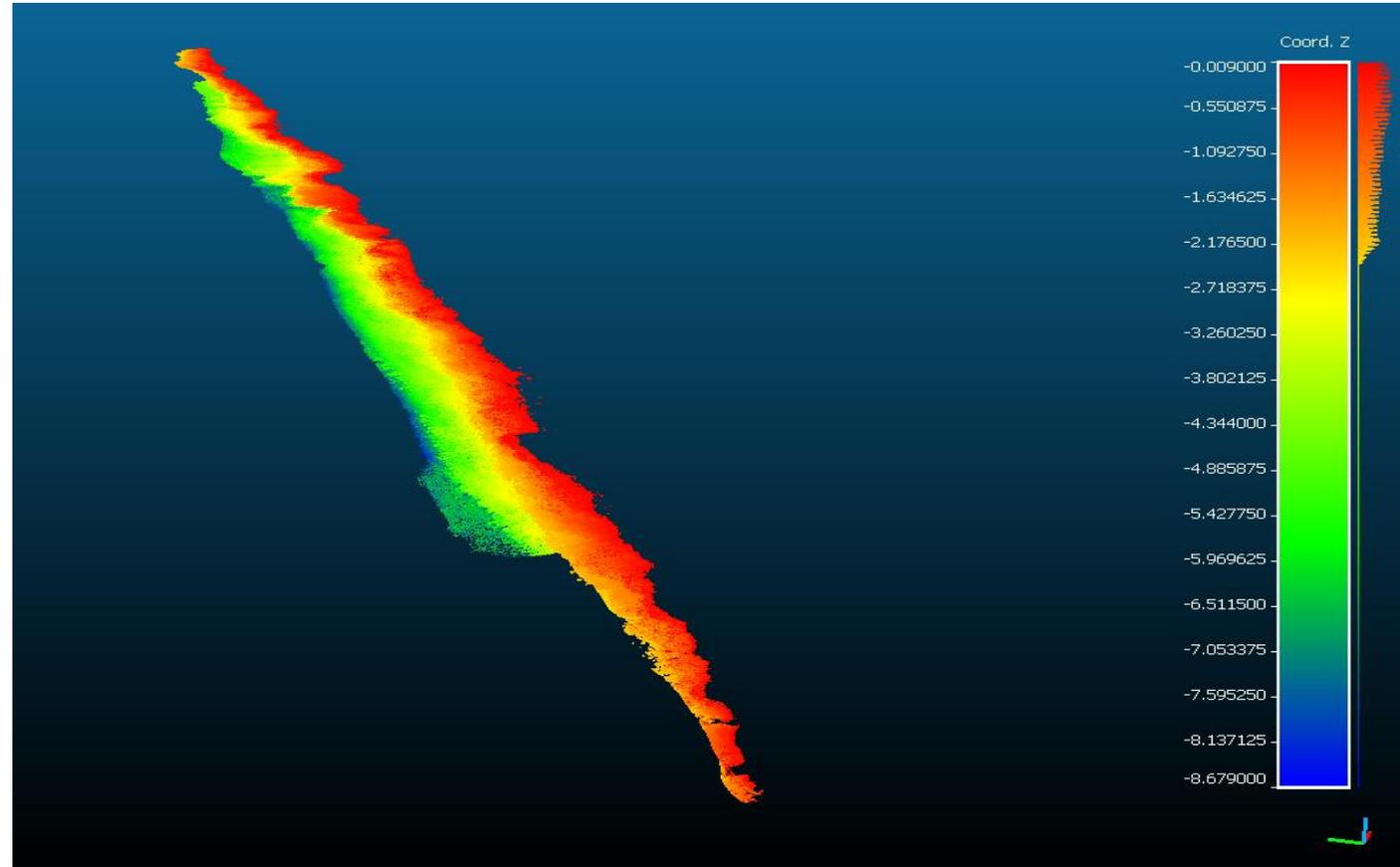


Extrait MNT .las de Fécamp : visualisation sous CloudCompare)

## Seuillage sous Python

Découpage et sélection du platier : code Python

Sélection du platier sous Python puis visualisation sous CloudCompare



Platier sélectionné pour les traitements : lecture sous CloudCompare)

## Détection automatique

Depuis CloudCompare



Export en images (png ou jpg)



Détection automatique de la fissuration du platier par seuillage et **filtre de Sobel**

**Code Python** interfacé avec Matlab.



Export en images géoréférencées

Fissuration platier  
1 – Détection  
automatique



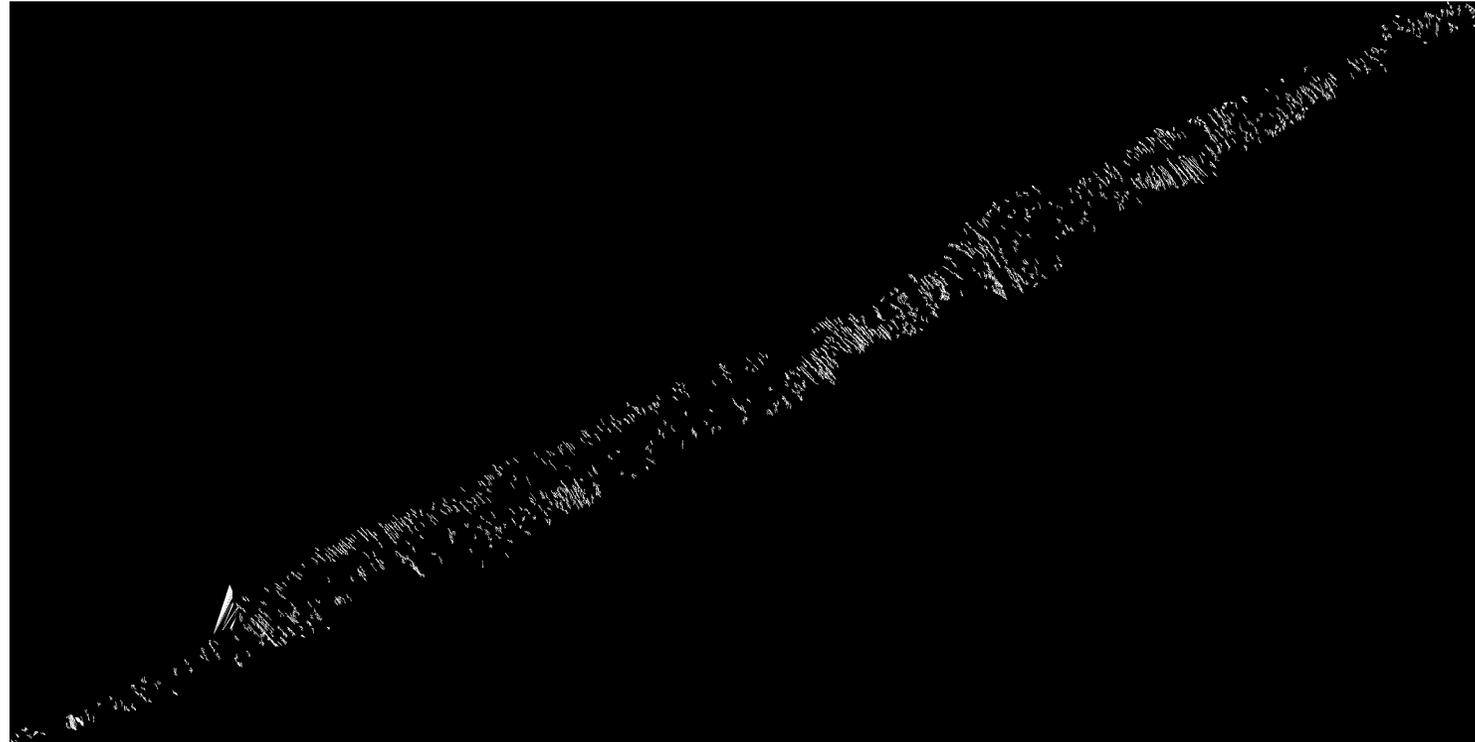
***Detection automatique : exemple de raster depuis CloudCompare***

Fissuration platier  
1 – Détection  
automatique



***Détection automatique : extraction fissures sur platier par filtre de Sobel (Matlab)***

Fissuration platier  
1 – Détection  
automatique



***Image des fissures géoréférencée : vers QGIS (shape)***

## Détection semi-automatique

Depuis CloudCompare



Export en images (ex. png) 600\*600 Px

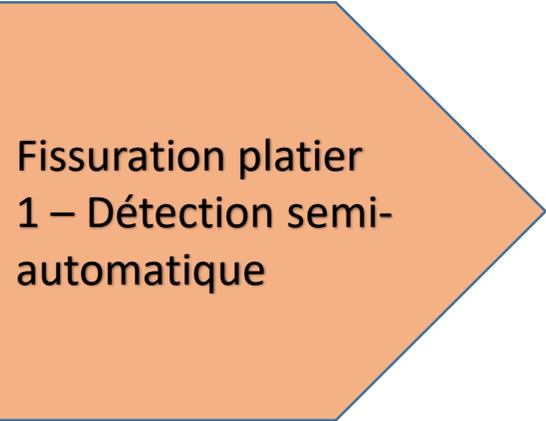


Détection semi-automatique de la fissuration du platier la méthode du **Fast-Marching**

**Code Python** interfacé avec Matlab (compilé en C++).



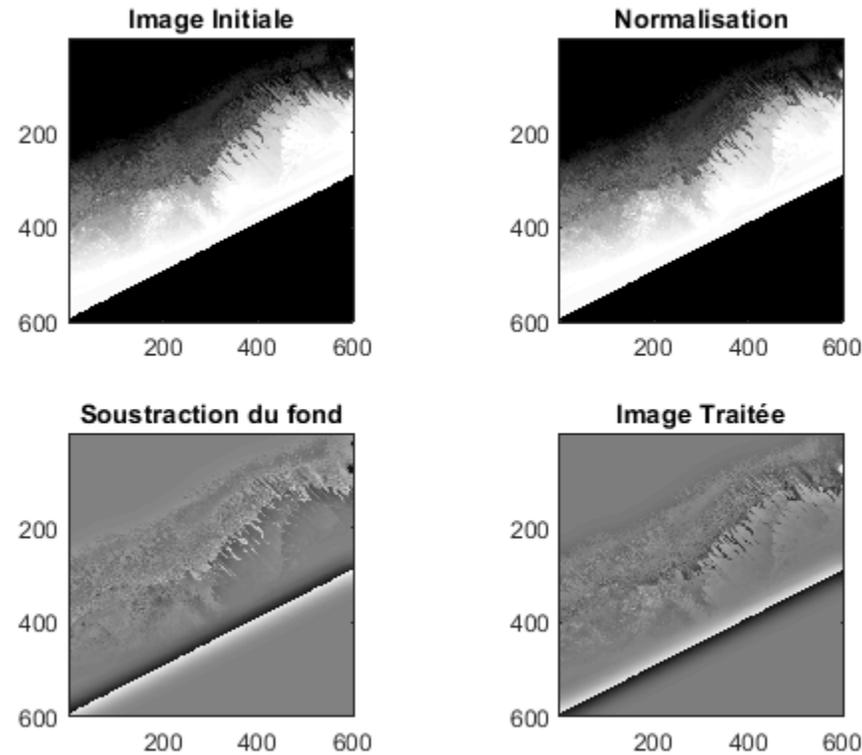
Export en images géoréférencées



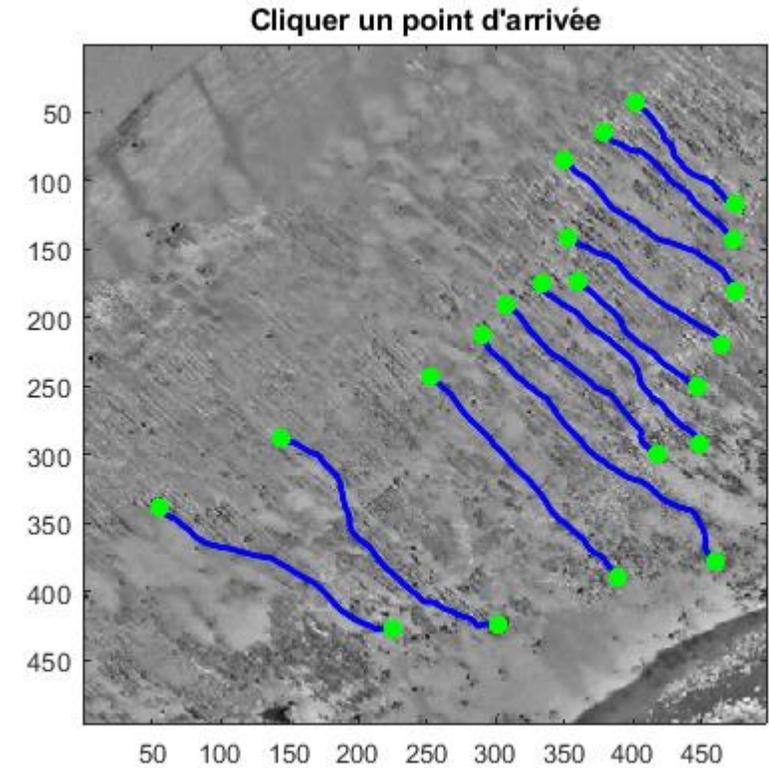
Fissuration platier  
1 – Détection semi-automatique

**Exemple de détection semi-automatique par Fast-Marching (code P. Charbonnier, Cerema/ENDSUM)**

Fissuration platier  
1 – Détection semi-automatique



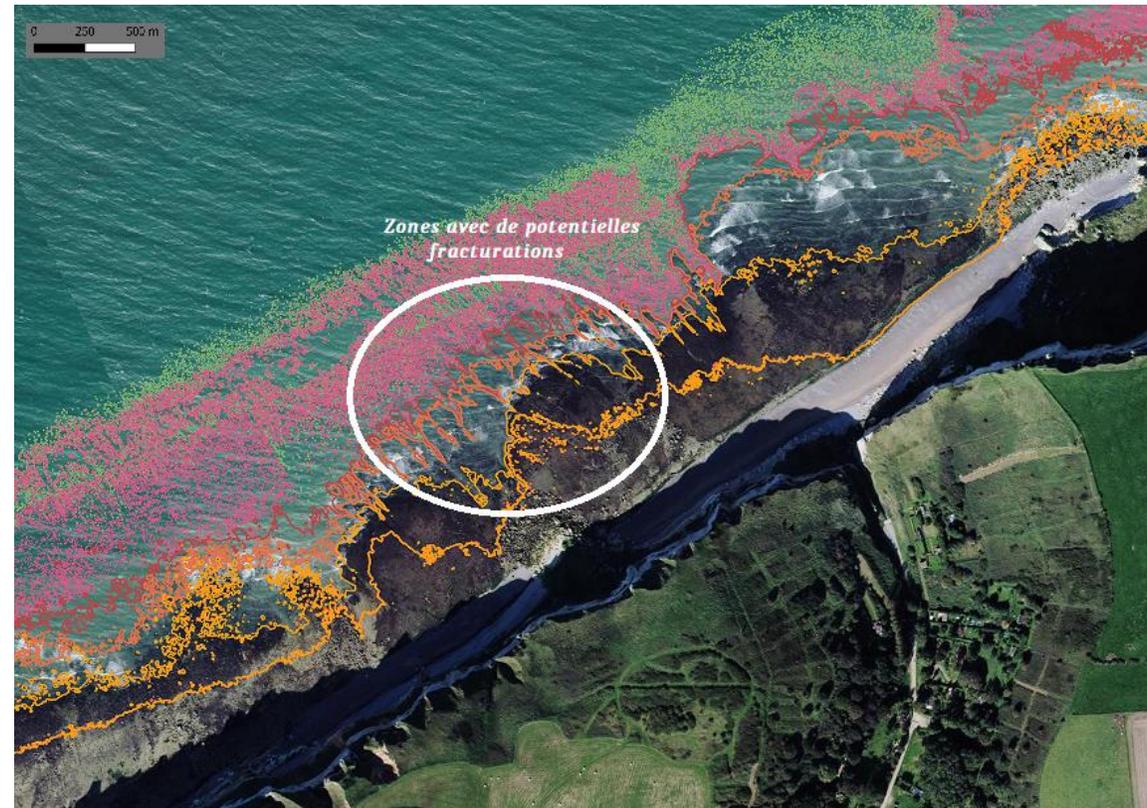
*Prétraitement (Matlab)*



*Sélection début/fin fissures*

## *Rendu sous SIG*

Export image + geotiff  
Visualisation sous SIG



*Exemple de seuillage (QGIS)*

## *Filtrage de Sobel et rendu sous SIG*

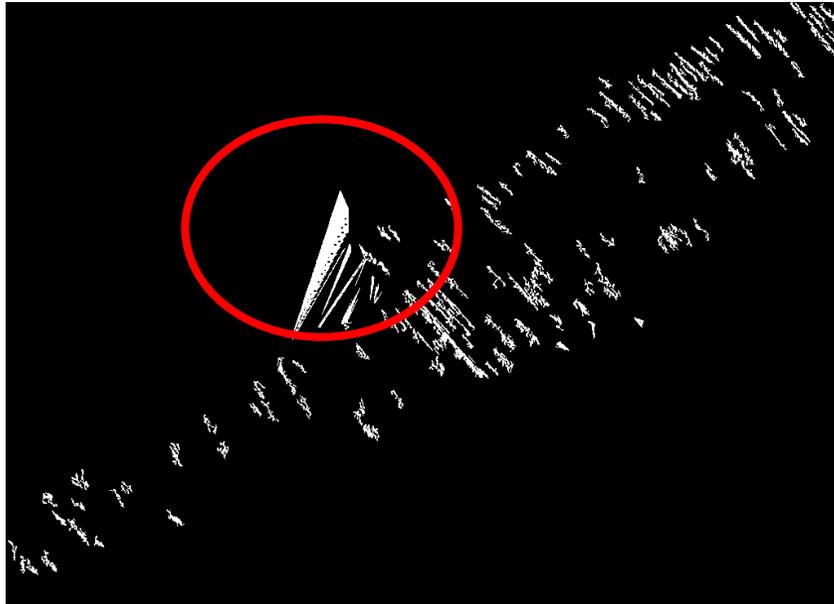


Export image + geotiff  
Visualisation sous SIG

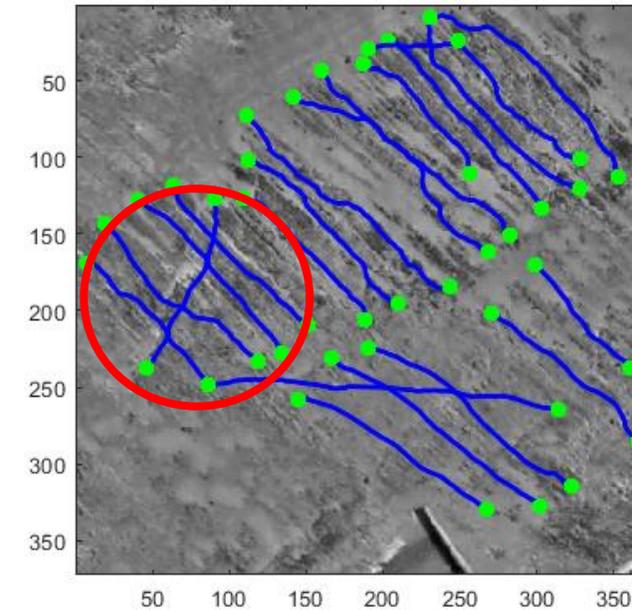
*Export des fissures sous QGIS (shape)*

## Limites

*Taille des images à traiter : limiter avec nos machines (50 Mo .txt pour méthode auto, 600\*600 Px/image pour méthode semi-auto)*



*Filtre de Sobel : réglage utilisateur empirique*



*Limites Fast Marching : que pointer?*

## Perspectives

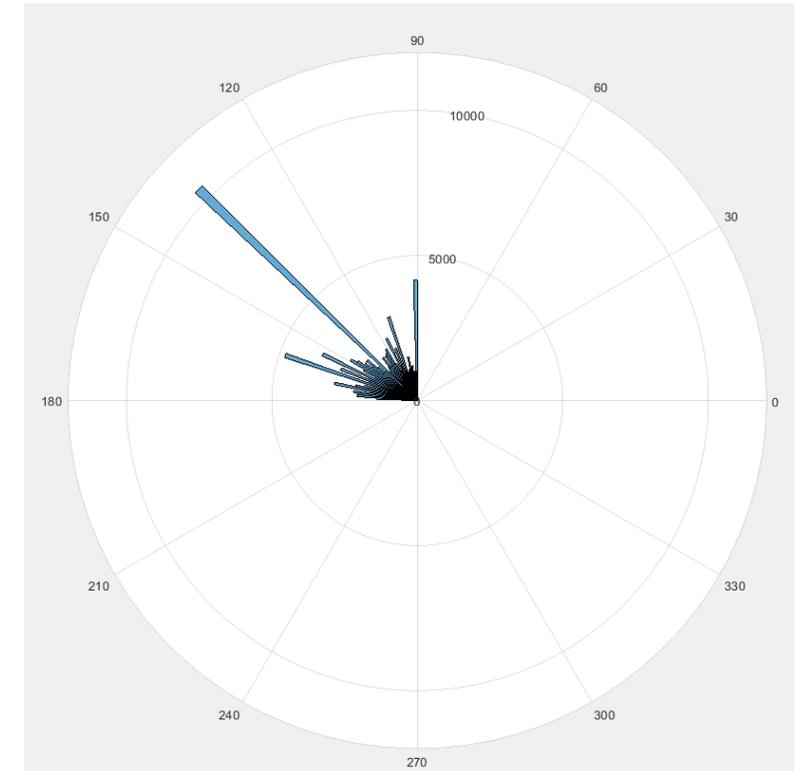
Automatiser les chaînes de traitements. Actuellement, passage par CloudCompare puis export en raster, puis traitement Matlab, puis QGIS =>

**Tout en Python**

Traiter de plus grandes images => **puissance de calcul**

Associer maintenant les **géologues pour la pertinence des relevés** de fissuration

Evaluer le degré de fissuration d'une falaise : relevé sur parois verticales, notamment falaises HN = vol drone



*Orientation des fissures à Fécamp (310°N)*



*Merci pour votre attention*