

# Détection et identification d'arbre à partir d'imagerie satellite/aérienne

Augustin Albert

30 avril 2021

## Table des matières

### Introduction

#### Position du problème

- pourquoi vouloir faire ça, utilité/contexte -1 pb extraction de données : différentes méthodes qui requièrent plus ou moins de matériel/images de qualité
- 2 pb traitement des données

#### État actuel de la recherche

- voir papiers

#### Objectifs du TIPE

- limitation à des images aériennes : pourquoi(moins coûteux, accessible sur internet, différents modes d'acquisitions, enjeux/difficultés -d'une part à concevoir ... pour détecter les a -d'une autre part à l'utiliser pour construire une base de données permettant l'identification ultérieure sur la base du machine learning
- application au site du parc régional...

## 1 Détection des houppiers

Intro : Méthode naïve ( les présentations) de détection des zones plus lumineuses. La luminosité des arbres peut beaucoup varier sur une même image (à moins d'avoir des images de haute qualité "prise en une seule fois" (ex papier). Une solution = Détection de blobs. 2 problèmes : -différence de luminosité et différence d'échelle. Citer algo pour. de pixels plus lumineux que leur voisins correspondants aux houppiers

- (a) 1  
(b) 2  
(c) 3

FIGURE 1 – test2

### 1.1 Laplacien du gaussien et approche multi échelle

Afin de prendre en compte des houppiers dont le diamètre peut varier considérablement entre les différentes espèces mais aussi au sein d'une même espèce, une approche multi-échelle est nécessaire. Nous utiliserons la théorie échelle-temps développée par (??indbergh). Une pyramide d'échelle est réalisée en lisant successivement l'image originelle avec un filtre gaussien de paramètre  $\sigma$  : le paramètre d'échelle. L'image originelle est convolué autant de fois que nécessaire par la fonction gaussienne suivante, le paramètre  $\sigma$  étant multiplié par un ratio fixé à!!!! important!!!! à chaque étape :

$$G_{\sigma} := \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

On applique alors un opérateur laplacien normalisé aux images résultante afin d'obtenir la pyramide d'échelle du laplacien du Gaussien (LoG). L'image obtenue à chaque niveau est alors la convolution de l'image originelle par la fonction Log :

$$LoG_{\sigma} := -\frac{1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

En pratique, le LoG est approximé par la différence du gaussien (DoG) obtenue en réalisant la difference des images floutées entre chaque niveau de la pyramide d'échelle.

Cette pyramide d'échelle du LoG permet d'extraire des caractéristiques indépendamment de leur échelle en exploitant la réponse de l'opérateur LoG appliqué à un signal échelon. (figure 5). Lorsque le rayon caractéristique du blob  $r$  varie, le minimum (maximum en valeur absolue) du LoG est atteint au centre du blob. Lorsque le paramètre  $\sigma$  varie, la réponse au centre est minimale lorsque  $r$  est relié à  $\sigma$  par la relation  $\sigma = \frac{r}{\sqrt{2}}$ . La réponse du LoG non normalisé s'atténuant lorsque  $\sigma$  augmente, l'opérateur a été multiplié par  $\sigma^2$  pour que la réponse soit indépendante de l'échelle.

La détection des blobs se ramène ainsi à la recherche d'un minimum local relativement à l'espace et global relativement à l'échelle pour identifier à la fois les centres des houppier et la taille caractéristique de leur rayon.

-figure 1 réponse à un echelon -figure 2 un blob mais différents paramètres de largeur et de parametre!!

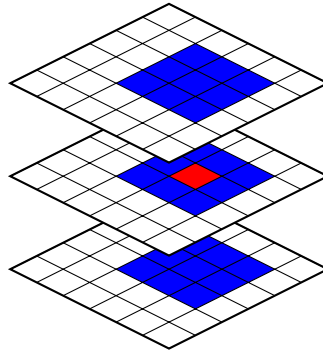


FIGURE 2 – test un deux

## 1.2 Mise en place de l'algorithme

L'algorithme envisagé à été implémenté à l'aide du langage Python et utilise la bibliothèque Numpy afin d'accélérer le traitement des tableaux. Il comporte trois étapes.

- L'image à traité est convertie en nuance de gris et éventuellement inversée afin que le fond soit plus clair que les houppiers. (une intervention humaine est nécessaire).

- La pyramide d'échelle du DoG est généré. Les noyaux de Gauss et la convolution ont été implémenté en exploitant la séparabilité du filtre de Gauss pour accélérer le calcul, mais en définitive, la convolution fourni par le module scipy à été utilisé. Les résultats sont stockés dans un tableau Numpy 3D.

- Les minimums sont calculés en parcourant le tableau par profondeur croissante afin de ne conserver que les houppiers de rayon maximum lors d'éventuels chevauchements.

3 paramètres permettent d'ajuster les résultats. La sélection des paramètres dépend fortement du (très heuristique comme méthode, nécessite des essais => la méthodes n'est pas complètement automatique. (dépend de la taille caractéristique des arbres, de l'échelle choisie) donner les paramètre pour l'échelle et tout.

facteur limitant i choix final des paramètres : + image utilisées (taille des arbres sur le terrain et échelle de l'image)

## 1.3 Évaluation des résultats

- évaluation de la complexité -propres résultats faire tableau 3 colonnes pour les trois images différentes -comparaison avec les résultats des papiers

## 2 Identification des espèces

Choix du machine learning comme dans de nombreux papiers (même si pas les mêmes données de base) Propre implémentation développé mais pas complet et ne permet pas à ce jour de traiter efficacement des images (manque le truc de convolution et reste lent en comparaison des systèmes professionnels) Le choix s'est donc porté sur Tensorflow.

### 2.1 Méthodologie de construction d'une base de données fiable

Il aurait été fastidieux d'étiqueter à la main un nombre important d'images, d'autant plus qu'elles sont de basses résolution. Afin - géoportail (vérifier autorisation... celle) et extraction sur des zones où la couverture d'espèce est uniforme : res — images triées en 2

### 2.2 entraînement et quelle type de modèle

-citer papier , à la main car ici le cas est plus simple mais l'on pourrait généraliser. (différence avec eux)

## 3 Prolongements envisageables

Différents prolongements seraient envisageables : -Sensible aux ... séparer préalablement et éventuellement grossièrement les zones forestières des zones d'habitation ou industrielle. Même une route bétonnée peut éventuellement altérer les résultats. batiement/construction qui fausserait les résultats. De plus obtient qu'un cercle autour des arbres -une méthode watershed segmentation avec marqueurs que l'on a trouvés pourrait être envisageable pour délimiter parfaitement les arbres (voir papier)

## A Compléments

Lien avec filtre basse fréquence (downsampling, pourquoi ?) le filtre de gauss retire les hautes fréquences donc permet d'augmenter la fréquence d'échantillonnage 'théorème de Shannon sans qu'il y ait d'artefacts.

## B Résultats

-résultats intermédiaire (pyramide de gauss) -les deux -douglas seul -feuillus seul -échantillon banque fourni pour feuillus -échantillon banque fourni pour douglas -échantillon aléatoire parmi des images non déjà vues

## **C algorithmes**

a faire : réaliser les tableaux de comparaison, mettre en forme le code écrire les preuves