

Rapport SAE Optimisation

SOMMAIRE

1. Introduction.....	2
2. Prétraitement de l'image.....	2
2.1 Méthodes de flou utilisées.....	2
2.2 Résultats du floutage.....	3
3. Détection des biomes.....	6
3.1 Méthodologie.....	6
3.2 Paramétrage et tests.....	6
3.3 Résultats.....	7
4. Étiquetage et affichage des biomes.....	7
4.1 Association clusters ↔ biomes.....	7
4.2 Visualisation.....	7
5. Détection des écosystèmes.....	8
5.1 Méthodologie.....	8
1. Algorithme choisi.....	8
2. Données utilisées.....	8
5.2 Résultats.....	8
6. Analyse des résultats.....	10

1. Introduction

Contexte scientifique : Une avancée scientifique majeure a été réalisée en juillet 2128, une nouvelle sorte de télescope a été inventée et permet désormais d'obtenir des images précises de surfaces de planètes très éloignées.

Il est donc possible de détecter la présence potentielle de formes de vie sur ces planètes en examinant les biomes qui s'y trouvent et leurs écosystèmes.

Objectif du projet : Notre équipe d'experts a donc été engagée pour développer une application détectant automatiquement les différents biomes et écosystèmes présents sur une planète à partir d'une image de la surface de celle-ci.

2. Prétraitement de l'image

2.1 Méthodes de flou utilisées

1. Filtre par moyenne

Nous avons utilisé une première méthode de floutage d'image simple qui consiste à appliquer des coefficients égaux à tous les pixels autour d'un pixel et d'appliquer la moyenne de couleur de ces pixels au pixel du milieu. Notre code permet d'utiliser une taille de kernel (matrice de pixels) variable dans le cadre du filtrage par moyenne.

2. Filtre gaussien

Le second filtre utilisé est un filtre Gaussien qui applique des coefficients plus élevés aux pixels les plus proches du pixel central. Cela permet d'utiliser des kernels plus grands tout en gardant un flou cohérent.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Nous avons testé des kernels de tailles 3x3 et 5x5 avec $\sigma \approx 1$ et 7x7 avec $\sigma \approx 1.5$.

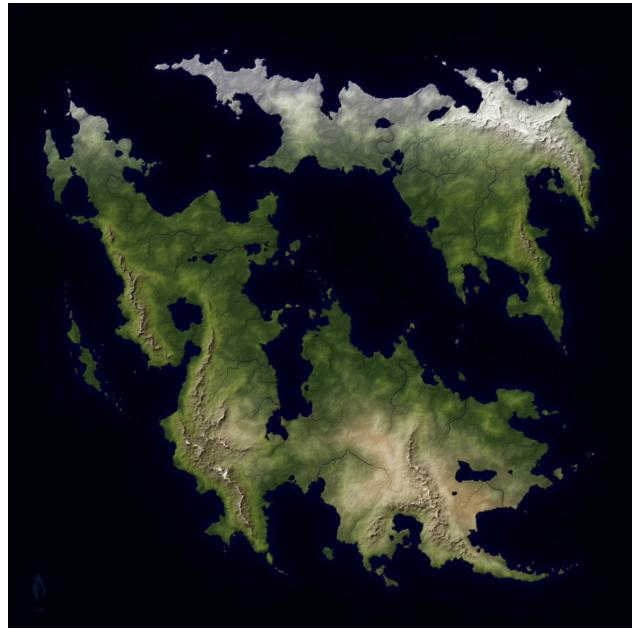
2.2 Résultats du floutage

1. Comparaisons

- a. Moyenne 3x3



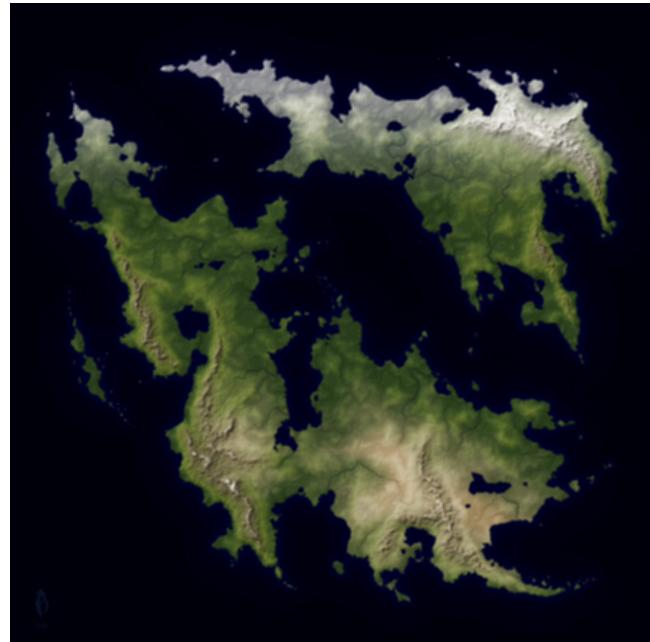
- b. Moyenne 5x5



c. Moyenne 7x7



d. Gaussien 3x3



e. Gaussien 5x5



f. Gaussien 7x7



2. Nous avons choisi d'utiliser le filtre de moyenne 5x5 car il permet un lissage efficace du bruit tout en conservant les contours des structures importantes.

3. Détection des biomes

3.1 Méthodologie

1. Algorithme choisi

Afin de développer une solution de reconnaissance de biomes, nous avons choisi d'utiliser l'algorithme K-Means car celui-ci a une faible complexité $O(N)$ et il correspond à ce dont on a besoin en proposant de déterminer à l'avance le nombre de clusters.

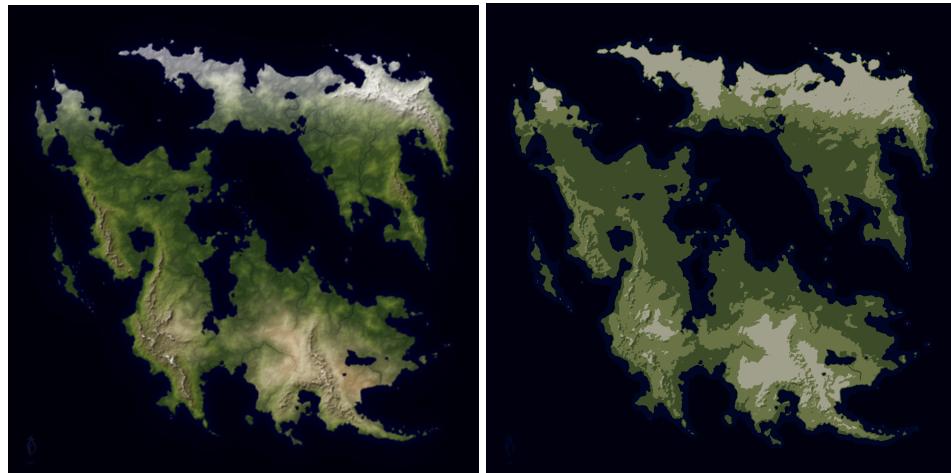
2. Seule l'information **RGB** des pixels est utilisée pour le clustering. Ce choix réduit la complexité du traitement tout en étant suffisant pour discriminer visuellement les types de biomes.

3.2 Paramétrage et tests

Le nombre de clusters a été testé entre 3 et 8.

Le choix final dépendait de la visualisation : on choisit un **k** qui permet de bien séparer les zones homogènes sans sur-segmentation. Nous avons donc choisi $k=5$.

3.3 Résultats



4. Étiquetage et affichage des biomes

4.1 Association clusters ↔ biomes

Le fichier Biome.java associe chaque cluster à un biome par une correspondance entre couleurs moyennes des clusters et une palette prédéfinie (PaletteData.java).

Une distance de couleur est utilisée pour associer un cluster à l'entrée la plus proche dans la palette.

4.2 Visualisation

Réalisée via BiomeDisplay.java, en superposant les couleurs de biomes détectés sur une version éclaircie de l'image originale (utilisation d'un fond clair).

Cela permet une lecture intuitive tout en gardant les formes originales visibles.

5. Détection des écosystèmes

5.1 Méthodologie

1. Algorithme choisi : DBSCAN2D.java

DBSCAN a été choisi pour détecter des regroupements spatiaux de pixels appartenant à un même biome. Il permet de détecter des formes non convexes et de gérer le bruit, ce qui est bien adapté à des structures naturelles.

2. Données utilisées :

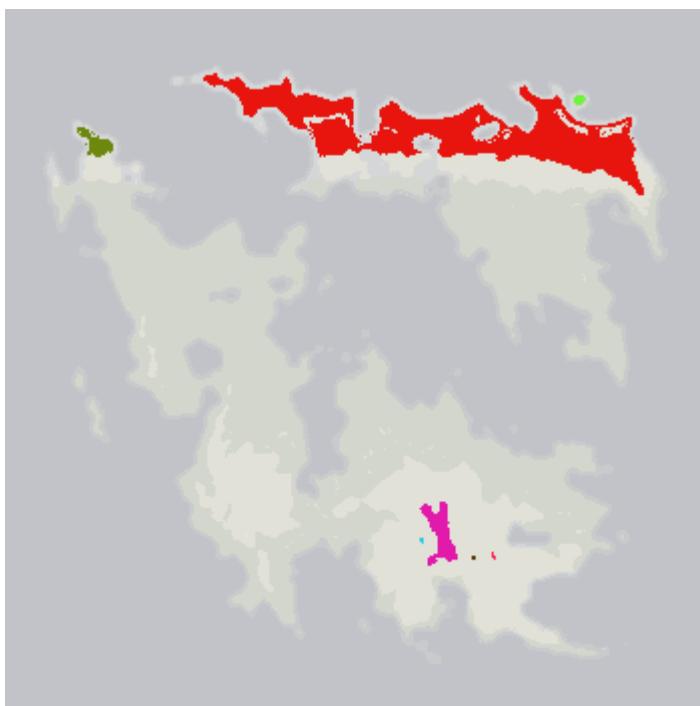
La position spatiale des pixels (coordonnées x, y).

La distance euclidienne est utilisée pour mesurer la proximité entre pixels.

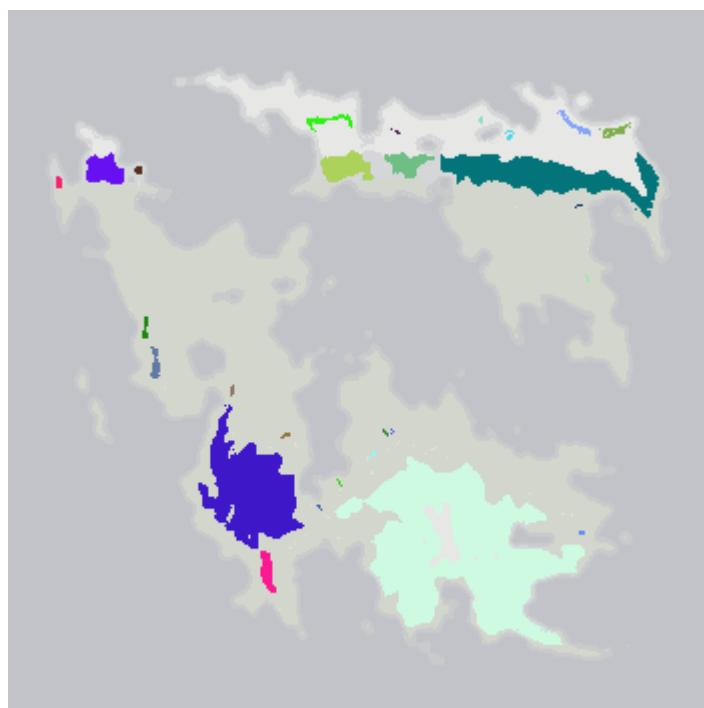
5.2 Résultats

Quelques exemples :

Glacier



Savane



Tundra



6. Analyse des résultats

Réussites du projet.

Segmentation claire des biomes.

Détection fine des regroupements spatiaux (écosystèmes).

Difficultés rencontrées

Floutage trop fort rendant les zones moins discriminables.

Choix du k dans K-Means plutôt difficile.

DBSCAN parfois trop sensible aux paramètres (epsilon, minPts).