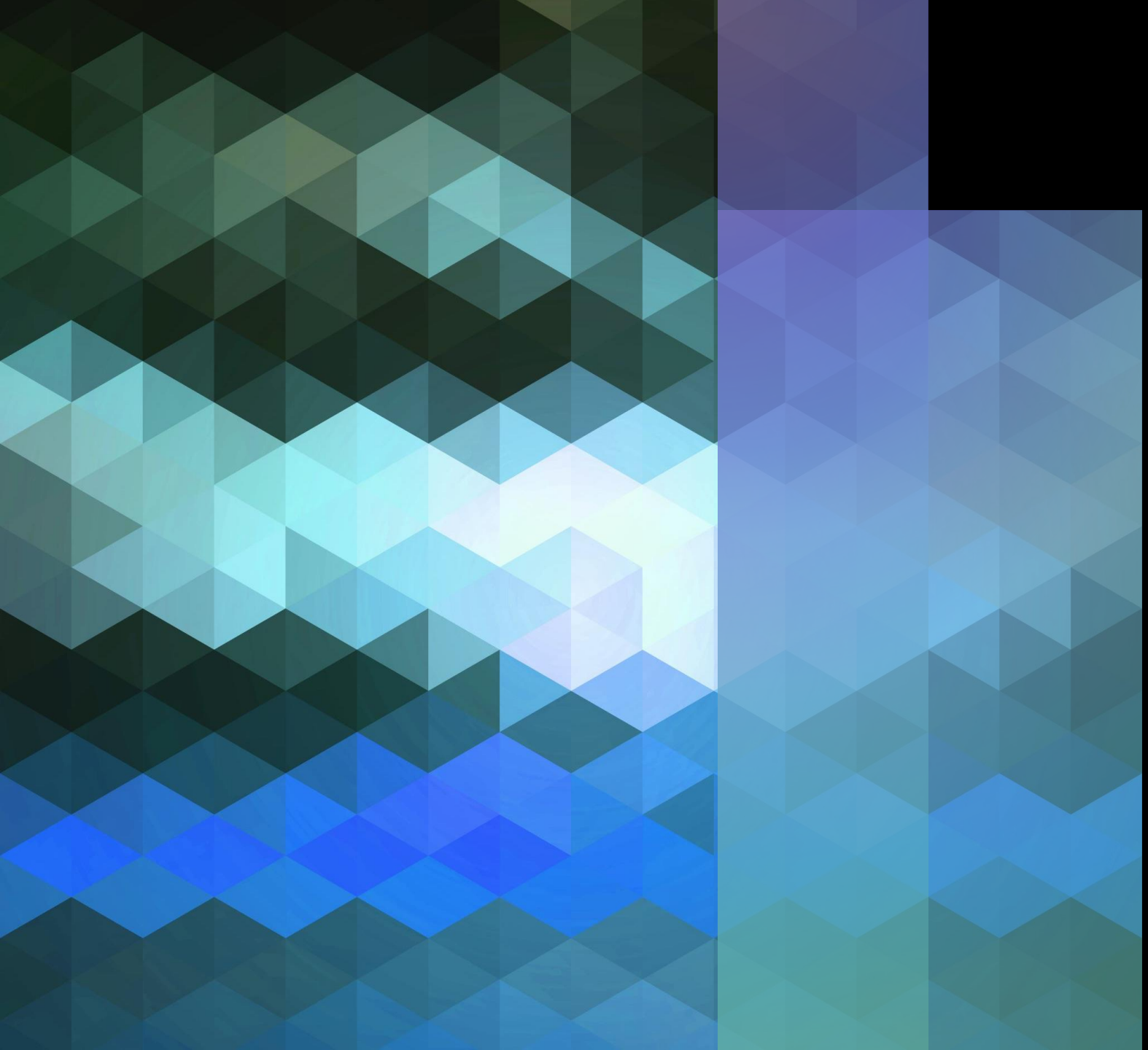


Region growing

Dorian SALMI

Mehdi GUITTARD

Structures

- Structure modélisant une région

```
struct Region {
    int id;
    vector<Point> content;
    vector<int> neighbors;
    float average;
    Vec3b region_color;
```

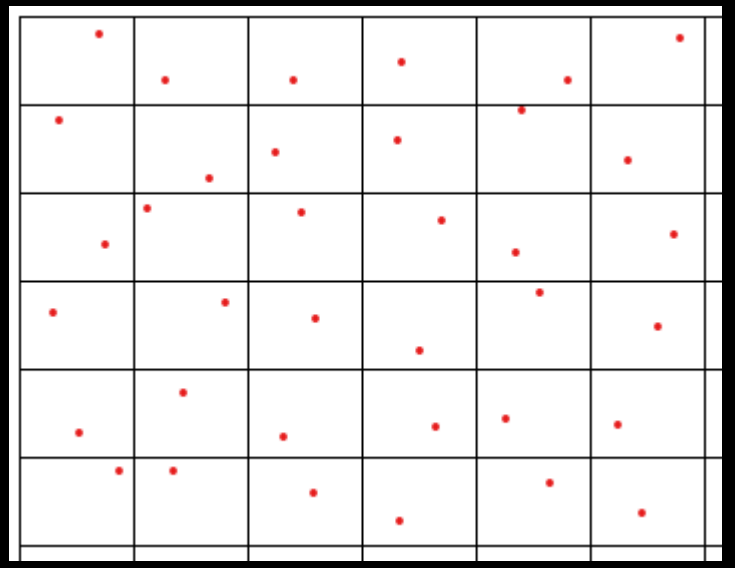
- Tableau 2D de couleurs (à la taille de l'image)

-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1

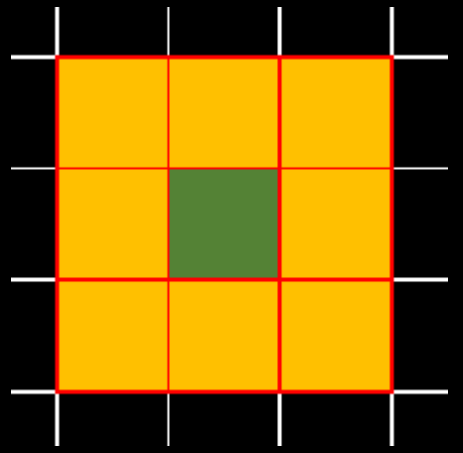
```
vector<Region> list_regions;
```

Seeds et parallèlisation

- Placement des graines



- Croissance des graines
  - Utilisation de threads



- Si l'image n'est pas entièrement couverte
  - Ajout de nouvelles graines

Critères d'inclusion et de fusion

- Deux thresholds différents
- Croissances des régions
  - Pixel à pixel

$$|\sqrt{(R_v - R_n)^2 + (G_v - G_n)^2 + (B_v - B_n)^2}| < threshold_{grow}$$

- Fusion des régions
  - Moyenne sur les régions

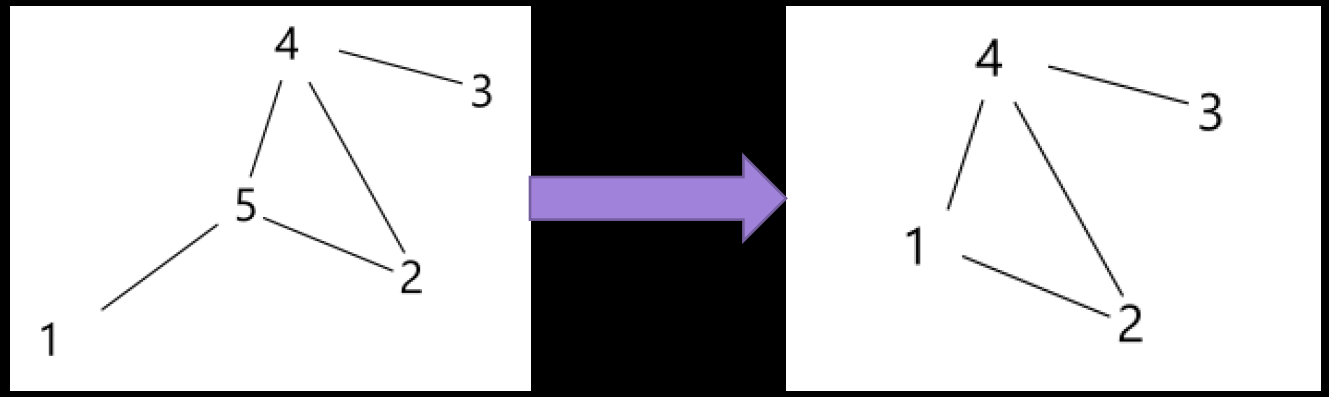
$$average_{pixel}(i) = \frac{R_i + G_i + B_i}{3}$$

$$average_{region} = \frac{\sum_{i=1}^{n} average_{pixel}(i)}{n}$$

$$average_{region1} - average_{region2} < threshold_{fusion}$$

Algorithme de fusion

- Graphe d'adjacence



- Méthode de fusion
  - Concaténation de listes

```
struct Region {
    int id;
    vector<Point> content;
    vector<int> neighbors;
    float average;
    Vec3b region_color;
```

- Parcours en profondeur
  - Moindre coût : fusionner des régions aussi grandes que possible

Démonstration

Merci de votre attention