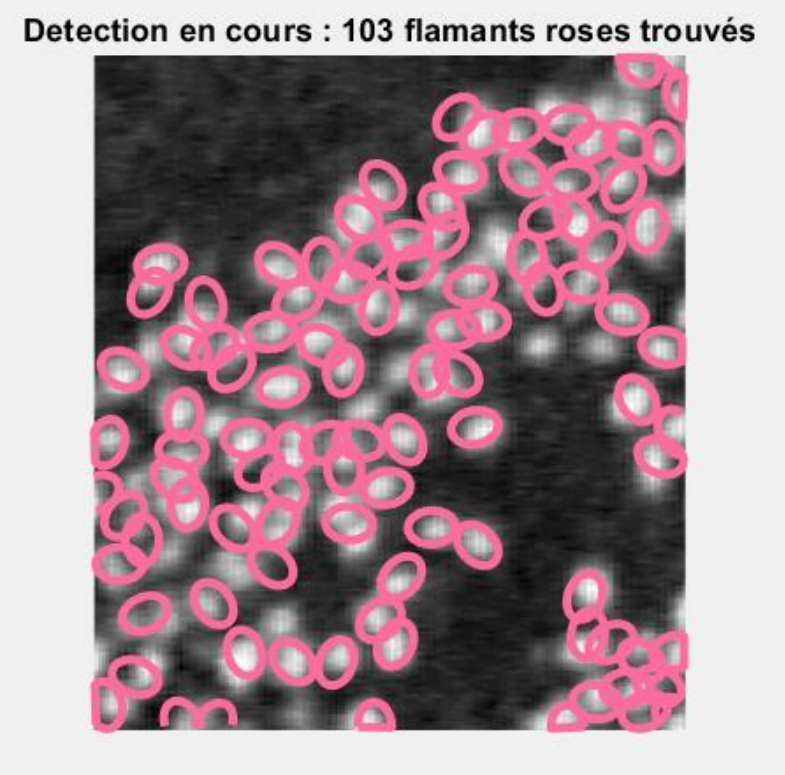

RAPPORT

STAGE IRIT

DARY Jean-Léo | Traitement des données Audio-Visuel | 01/06/21 – 13/08/21

Table des matières

Détection de flamants roses par des ellipses

Rappel du problème

Nous voulons minimiser le champ de Markov de formule suivante :

$$U(c) = -\sum_{1\le i\le N} \bar{I}(c_i) + \beta \sum_{1<i<j\le N} \delta(\|c_j - c_i\| \le \sqrt{2}\ R)$$

Nous reformulons le problème grâce au recuit simulé par la méthode suivante :

$$U(c) = -\sum_{1\le i\le N} U_i(c_i) + \beta \sum_{1<i<j\le N} \delta(\|c_j - c_i\| \le \sqrt{2}\ R)$$

Où :

$$U_i(c_i) = 1 - \frac{2}{1 + \exp\left(-\gamma\left[\frac{\bar{\bar{I}}(c_i)}{s} - 1\right]\right)}$$

1) DETECTION CERLCE FIXE

Pour cette première partie nous avons fait m'états des algorithmes déjà existant a savoir la méthode décrite ci-dessus avec des cercle de rayon R. Ce sont les fichier *Cercle_detection_fixe_geometry.m, Cercle_detection_fixe_No_geometry.m.* Ces algorithmes sont pour détecter un nombre fixe d'objets, il a donc été introduit un algorithme de naissances/morts multiples. Le principe de cet algorithme est d'alterner les phases de naissances, où de nouveaux disques sont ajoutés aléatoirement à la configuration courante, et les phases de morts, où les disques les moins pertinents, au sens de l'énergie, sont supprimés. Cette dynamique, intégrée dans un schéma de recuit simulé, converge vers le minimum global de l'énergie. Pour ainsi avoir un nombre de cercle variable. (Fichier *Cercle_detection_var.m*)

2) DETECTION ELLIPSE FIXE

Nous avons choisi de modéliser des ellipses par des ellipse fixe ou $a = R$ et $b = \sqrt{2}\,R$ Nous avons pour cela changer le calcul de $\bar{I}(c_i)$. Pour une ellipse l'équation cartésienne devient: $\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 \le 1$ on ne prend plus que les points qui vérifie cette équation dans le calcul de $\bar{I}(c_i)$. . Pour la distance entre 2 ellipses qui intervient dans le terme de droite nous avons approximer en première approche la distance entre 2 cercles pour 2 ellipses (ce qui est bien sur totalement faux). Le terme dans l'énergie pour le recouvrement entre ellipse est assimilé à un cercle de rayon R_recouvrement = b.

3) DETECTION ELLIPSE FIXE V2

Nous ne sommes donc intéressées à la distance entre ellipses. Ce qui n'est pas du tout évidents et peu être couteux en tant de calculs. Pour éviter d'être trop long en temps d'exécution nous avons choisi une approche discrète du problème. En effet nos ellipses de taille fixe sont discrétisées en un certain nombre de points (*nb_points_disque)*.

L'idée est de compter le nombre de point dans l'intersection des 2 ellipses ou mieux reformuler : combien de point de l'ellipse 1 se situe à l'intérieur de l'ellipse 2 ou combien de point de l'ellipse 2 se situe à l'intérieur de l'ellipse 1. Pour cela nous avons utilisé la méthode suivante : On définit la matrice de projection :

$$T = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & xc \\ \sin(\theta) & \cos(\theta) & yc \\ 0 & 0 & 1 \end{bmatrix}$$

On appelle matrice ellipse la matrice définit par (codé dans *param2ellipse.m*) :

$$C = (T^{-1})^t \begin{bmatrix} \frac{1}{a^2} & 0 & 0 \\ 0 & \frac{1}{b^2} & 0 \\ 0 & 0 & -1 \end{bmatrix} T^{-1}$$

Cette matrice à la particularité que quand on le calcul de la norme d'un point de $\mathbb{R}^3$ induite par le produit scalaire issue de C on a $\|x\|_C = x^t\, C\, x$ si $\|x\|_C > 0$ alors le point est en dehors de l'ellipse et si $\|x\|_C < 0$ le point est à l'intérieur de l'ellipse. De ce constat pour calculer le croisement entre 2 ellipses. On calcule

$$croissement(ellipse\ 1)_{ellipse\ 2} = \sum_{Points_i \in \{ellipse\ 1\}} \delta(\|Points\|_{C_{ellipse\ 2}} < 0)$$

Pour que cette relation soit symétrique on rajoute à cette équation :

$$croissement(ellipse1, ellipse2) = \max\left(croissement(ellipse\ 1)_{ellipse\ 2}, croissement(ellipse\ 2)_{ellipse\ 1}\right)$$

$croissement(ellipse\ 1)_{ellipse\ 2}$ a été codé dans *distance_ellipse* ensuite pour avoir le même fonctionnement que l'algorithme précédent on établit heuristiquement un seuil :

$$distmax = nbPointsDisques * PourcentageEllipseCroisé$$

Où on a établi un chevauchement qui « marche bien » :

$$PourcentageEllipseCroisé = 25\%$$

Ainsi la formule précédente devient :

$$U(c) = -\sum_{1 \le i \le N} U_i(c_i) + \beta \sum_{1 < i < j \le N} \delta(croissement(ellipse_j, ellipse_i) \le distmax)$$

Tout est codé dans *Ellipse_fixe_v2.m*

4) DETECTION ELLIPSE V2 OPTI

Les performances de cet algorithme ne sont pas optimisées nous avons donc décider d'améliorer cet algorithme en précalculant les résultats de la fonction $croissement(ellipse_j, ellipse_i)$. Pour cela nous avons discrétiser les valeurs possibles

d'ellipse. On discrétise θ et x_c, y_c sachant que $\theta \in [0,2\pi]$ puisque on pose la première ellipse à $\theta = 0$, et la 2 -ème ellipse à $\theta = |\theta_1 - \theta_2|$ et $x_c, y_c \in [-2b, 2b]$ puisque on place la première ellipse à [0,0] et la seconde à $[x_{c1} - x_{c2}, y_{c1} - y_{c2}]$ et donc si l'écart est plus grand que 2b, l'ellipse ne se croise forcement pas donc pas besoin de calculer le croissement, cette discrétisation est paramétrable dans le fichier *discretisation.m*. Ensuite il suffit de calculer l'interpolation au plus proche voisin lorsque l'on veut calculer la distance :

$$indice_x = round\left(\frac{(x_1 - x_2) - x_{min}}{x_{max} - x_{min}}\, nx\right) + 1$$

$$indice_y = round\left(\frac{(y_1 - y2) - x_{min}}{y_{max} - x_{min}}\, ny\right) + 1$$

$$indice_\theta = round\left(\frac{|\theta_1 - \theta_2| - \theta_{min}}{\theta_{max} - \theta_{min}}\, n\theta\right) + 1$$

Où $nx\ , ny, n\theta$ est le nombre de point de la discrétisation pour chaque axe et $x_{min/max}, y_{min/max}, \theta_{min/max}$ sont les limites de la discrétisation de chaque axe. Il suffit ensuite d'évaluer la distance avec $f(indice_x, indice_y, indice_\theta)$ (Res dans le fichier *Ellipse_fixe_v2_Opti.m*)

Collage