TP4 – Estimation des paramètres d'une ellipse

À partir d'un ensemble de points dont la forme rappelle celle d'une ellipse, est-il possible de trouver l'ellipse qui passe « au plus près » de ces points ? Ce genre de question se pose très souvent en analyse d'images. Il s'agit d'un problème d'*estimation*. Dans ce TP, vous allez tester plusieurs méthodes d'estimation des paramètres d'une ellipse. Vous utiliserez l'une d'entre elles dans une application de réalité augmentée.

Commencez par lancer le script `donnees.m`, qui affiche une ellipse tirée aléatoirement, ainsi que n points placés aléatoirement sur cette ellipse, dont chaque coordonnée est bruitée par un bruit additif gaussien.

Exercice 1 : estimation par le maximum de vraisemblance

Le maximum de vraisemblance est une technique très générale d'estimation de paramètres, qui a déjà été illustrée dans plusieurs TP de Probabilités/Statistiques en 1A.

Si $(X_1, \ldots, X_n)$ est un n-uplet de variables aléatoires continues « iid » (indépendantes et identiquement distribuées) de même densité de probabilité $f_p(x)$ dépendant d'une liste de paramètres formant un vecteur p, alors la probabilité pour que n observations $(x_1, \ldots, x_n)$ constituent une réalisation de ce n-uplet est appelée *vraisemblance* (souvent notée L, pour *Likelihood*). Grâce au caractère « iid » des variables aléatoires, elle s'écrit :

$$L_p(x_1, \ldots, x_n) = \prod_{i=1}^{n} f_p(x_i) \tag{1}$$

Une façon de caractériser une ellipse consiste à choisir deux points du plan F_1 et F_2, appelés *foyers*, et un réel a tel que $2a > 2c = d(F_1, F_2)$, où $d(\cdot,\cdot)$ désigne la distance euclidienne. L'ensemble des points P tels que $d(P, F_1) + d(P, F_2) = 2a$ constitue une ellipse (cette définition de l'ellipse explique son tracé par la méthode dite « du jardinier »). On dispose donc d'un critère permettant d'évaluer si un point P du plan se trouve plus ou moins près de l'ellipse. En notant $e(P) = d(P, F_1) + d(P, F_2) - 2a$ l'écart entre les deux membres de l'équation précédente, il est légitime de modéliser ces écarts par une **loi normale tronquée** :

$$f_p(P) = \begin{cases} K \, \exp\left\{-\frac{e(P)^2}{2\,\sigma^2}\right\} & \text{si } e(P) \geq \min\limits_{P \in \mathbb{R}^2}\{e(P)\} = d(F_1, F_2) - 2a = 2(c-a) < 0 \\ 0 \quad \text{sinon} \end{cases} \tag{2}$$

où $p = (F_1, F_2, a, \sigma)$. L'écart $e(P)$ prenant ses valeurs dans $[2(c-a), +\infty[$ et non dans $\mathbb{R}$, le coefficient de normalisation K n'est pas égal à $1/(\sigma\sqrt{2\pi})$. Néanmoins, si $2|c-a| \gg \sigma$, on montre que $K \approx 1/(\sigma\sqrt{2\pi})$.

L'*estimation par maximum de vraisemblance* consiste à chercher la valeur $\widehat{p}$ de p qui maximise la vraisemblance des n points du plan $(P_1, \ldots, P_n)$. Comme un produit est plus difficile à maximiser qu'une somme, et que la fonction logarithme est strictement croissante, il est préférable de maximiser la *log-vraisemblance* :

$$\widehat{p} = \underset{p \in (\mathbb{R}^2)^2 \times (\mathbb{R}^+)^2}{\arg\max} \left\{ \ln \left[\prod_{i=1}^{n} f_p(P_i) \right] \right\} = \underset{p \in (\mathbb{R}^2)^2 \times (\mathbb{R}^+)^2}{\arg\max} \left\{ n \ln K - \frac{1}{2\,\sigma^2} \sum_{i=1}^{n} e(P_i)^2 \right\} \tag{3}$$

Parmi les paramètres de la loi, l'écart-type σ n'est pas caractéristique de l'ellipse. Si nous supposons que ce paramètre a une valeur suffisamment faible pour que la condition $2|c-a| \gg \sigma$ soit vérifiée, alors K ne dépend que de σ. Cela permet de simplifier le problème (3) :

$$(\widehat{F_1}, \widehat{F_2}, \widehat{a}) = \underset{(F_1, F_2, a) \in (\mathbb{R}^2)^2 \times \mathbb{R}^+}{\arg\max} \left\{ -\sum_{i=1}^{n} e(P_i)^2 \right\} = \underset{(F_1, F_2, a) \in (\mathbb{R}^2)^2 \times \mathbb{R}^+}{\arg\min} \left\{ \sum_{i=1}^{n} e(P_i)^2 \right\} \tag{4}$$

Complétez le script `exercice_1.m` par l'écriture de la fonction `MV`, qui résout le problème (4) par tirages aléatoires selon des lois uniformes. Le résultat est évalué par son score $(A_0 \cap \widehat{A})/(A_0 \cup \widehat{A}) \in [0, 1]$, où A_0 désigne l'aire de l'ellipse originale et $\widehat{A}$ celle de l'ellipse estimée.

Exercice 2 : estimation par les moindres carrés ordinaires

L'équation cartésienne d'une conique (ellipse, parabole, hyperbole) est une équation polynomiale de degré 2 :

$$\alpha\,x^2 + \beta\,x\,y + \gamma\,y^2 + \delta\,x + \epsilon\,y + \phi = 0 \tag{5}$$

Les six paramètres $(\alpha, \beta, \gamma, \delta, \epsilon, \phi)$ ne sont pas indépendants, puisque la conique ne change pas si ces six paramètres sont multipliés par un même coefficient. Cela signifie qu'une conique possède cinq « degrés de liberté », ce qui est cohérent avec le modèle d'ellipse de l'exercice 1, de paramètres (F_1, F_2, a).

Les équations (5) forment un système linéaire *homogène* $A\,X = 0$, où $X = [\alpha, \beta, \gamma, \delta, \epsilon, \phi]^\top$ et A est une matrice de taille $n \times 6$. Pour éviter la solution « triviale » $\widehat{X} = 0$, on peut imposer une contrainte linéaire sur les paramètres, par exemple $\alpha + \gamma = 1$ (comme une ellipse vérifie $\beta^2 - 4\alpha\gamma < 0$, il est impossible que $\alpha + \gamma = 0$). Cette contrainte ajoute une équation linéaire *non homogène* au système des équations (5), qui n'admet donc plus la solution triviale $\widehat{X} = 0$. Complétez le script `exercice_2.m` par l'écriture de la fonction `MCO`, qui estime X par résolution approchée du nouveau système, au sens des moindres carrés ordinaires (cf. TP3). Vous constatez que l'estimation est bien plus fiable par les moindres carrés ordinaires que par le maximum de vraisemblance.

Application à la réalité augmentée

Le répertoire `images_tag` contient une séquence d'images réelles d'un livre posé sur une table, près duquel se trouve une cible plane composée de cercles concentriques. Or, la projection perspective d'un cercle sur un plan (ici, le plan image de l'appareil photographique) est une ellipse. La méthode de détection de l'exercice 2 permet donc d'estimer les différentes ellipses, et d'en déduire la position de la caméra à chaque prise de vue.

Lancez le script `augmentation.m`, qui effectue l'*augmentation* de cette séquence d'images réelles par des images de flammes du TP3. Une fois le calcul terminé, qui est relativement long, lancez le script `cinema.m`.

Exercice 3 : estimation par les moindres carrés totaux

Les abscisses x_i et les ordonnées y_i des points P_i étant simultanément bruitées, il semble légitime d'estimer les paramètres de l'ellipse par les *moindres carrés totaux* (cf. l'estimation de la droite de régression vue en cours). Cela revient à résoudre les équations (5) de manière approchée, sous la contrainte non linéaire $\|X\| = 1$:

$$\widehat{X} = \underset{\|X\|=1}{\arg\min} \left\{ \|A\,X\|^2 \right\} \tag{6}$$

La résolution d'un tel problème se fait en introduisant un *lagrangien* (λ est un *multiplicateur de Lagrange*) :

$$\mathcal{L}(X, \lambda) = \|A\,X\|^2 + \lambda\,(1 - \|X\|^2) \tag{7}$$

La condition d'optimalité $\nabla\mathcal{L} = 0$ donne les deux équations suivantes :

$$\left\{ \begin{array}{l} \nabla_X \mathcal{L}(X, \lambda) = 0 \\ \nabla_\lambda \mathcal{L}(X, \lambda) = 0 \end{array} \right. \iff \left\{ \begin{array}{l} A^\top A\,X = \lambda\,X \\ \|X\| = 1 \end{array} \right. \tag{8}$$

La solution du problème (6) est donc un vecteur propre de $A^\top A$ de norme 1. Or, cette matrice est symétrique, réelle. Elle admet donc une base orthonormée de vecteurs propres. Comme $\|A\,X\|^2 = X^\top A^\top A\,X = \lambda\,X^\top X = \lambda$, et que les valeurs propres de $A^\top A$ sont toutes positives ou nulles, puisque cette matrice est semi-définie positive, on en conclut que la solution $\widehat{X}$ de (6) est un vecteur propre de $A^\top A$ associé à sa plus petite valeur propre. Cette solution s'obtient également par *décomposition en valeurs singulières* de A (`[U,S,V] = svd(A)` en Matlab) : $\widehat{X}$ est alors égal à la dernière colonne de V, si les valeurs singulières sont triées par ordre décroissant (cf. cours).

Faites une copie du script `exercice_2.m`, de nom `exercice_3.m`, que vous modifierez de manière à tester cette troisième méthode d'estimation. Vous constatez que les résultats sont nettement moins bons qu'avec la méthode précédente. Cela vient de ce que le bruit sur les éléments de la matrice A n'est pas gaussien, puisque ces éléments sont de degrés 0, 1 ou 2 en x_i et y_i. En centrant et en normalisant judicieusement les données, montrez qu'il est toutefois possible d'améliorer le résultat de cette estimation.