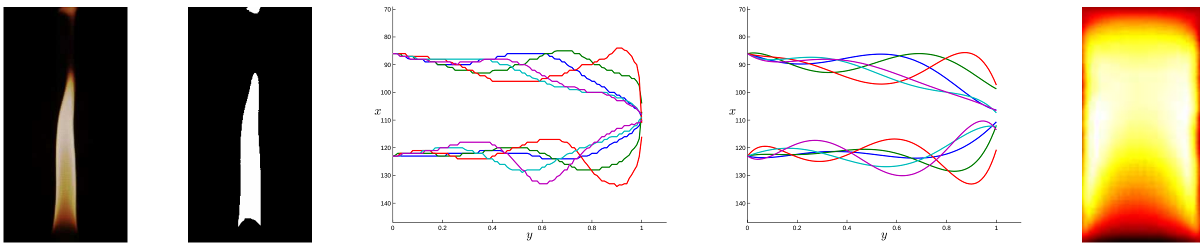
TP3 – Simulation d'une flamme de bougie

On souhaite simuler, de la façon la plus réaliste possible, une séquence d'images d'une flamme de bougie. La figure 1 illustre les prétraitements effectués sur une séquence réelle de n images : un seuillage appliqué à chaque image (cf. figure 1-a) fournit une image binaire (cf. figure 1-b), sur laquelle la silhouette de la flamme est détectée. Après normalisation, toutes ces silhouettes ont une même hauteur égale à 1. En tournant les axes d'un quart de tour vers la droite, les abscisses x sont orientées vers le bas et les ordonnées y vers la droite (cf. figure 1-c). Lancez le script `donnees.m`, afin de créer la matrice tridimensionnelle `bords`, de taille $p \times 2 \times n$, telle que `bords(j,1,k)` et `bords(j,2,k)` contiennent les abscisses des bords gauche et droit de la $k^{\text{ème}}$ silhouette (devenus ses bords supérieur et inférieur, après rotation d'un quart de tour), à l'ordonnée $y = (j-1)/(p-1)$, $j \in [1, p]$. La figure 1-c montre que les silhouettes ont toutes la même base, c'est-à-dire que `bords(1,1,k)` et `bords(1,2,k)` ne dépendent pas de `k` (et valent, respectivement, 86 et 123).



(a) (b) (c) (d) (e)

FIGURE 1 – (a) Une des images de la séquence réelle. (b) Détection de la silhouette par seuillage. (c) Silhouettes de la séquence réelle, après normalisation et rotation. (d) Modélisation des silhouettes par des paires de courbes de Bézier indépendantes. (e) Texture moyenne normalisée, calculée à partir des n images originales.

Utilisation de courbes de Bézier pour la modélisation des silhouettes

Chaque bord de chaque silhouette peut être modélisé par une courbe de Bézier de degré d, entièrement définie par $d+1$ points de contrôle, dont les ordonnées $\alpha_i = i/d$, $i \in [0, d]$, sont équiréparties dans l'intervalle $[0, 1]$. Les points de contrôle sont notés $P_i^k = (\beta_i^k, \alpha_i)$ pour le bord gauche (supérieur) de la $k^{\text{ème}}$ silhouette, et $Q_i^k = (\gamma_i^k, \alpha_i)$ pour son bord droit (inférieur). Les abscisses $\beta_0^k = 86$ et $\gamma_0^k = 123$ étant indépendantes de k, sont notées β_0 et γ_0. Les bords de la silhouette numéro k sont donc modélisés par les équations :

$$\left\{ \begin{array}{l} x = \beta_0 \, B_0^d(y) + \displaystyle\sum_{i=1}^{d} \beta_i^k \, B_i^d(y) \\ x = \gamma_0 \, B_0^d(y) + \displaystyle\sum_{i=1}^{d} \gamma_i^k \, B_i^d(y) \end{array} \right.$$

où les fonctions $B_i^d(y) = \mathrm{C}_d^i \, y^i \, (1-y)^{d-i}$, $i \in [0, d]$, sont les *polynômes de Bernstein de degré d*. Dans la fonction `bezier`, vous remarquez que C_d^i (« nombre de combinaisons ») s'écrit `nchoosek(d,i)` en Matlab.

Modéliser la silhouette numéro k consiste donc à estimer les d paramètres $(\beta_1^k, \ldots, \beta_d^k)$ de son bord gauche et les d paramètres $(\gamma_1^k, \ldots, \gamma_d^k)$ de son bord droit. La figure 1-d montre les modélisations obtenues à partir des silhouettes de la figure 1-c. Il est notable que les sommets des flammes ne sont pas fermés.

Exercice 1 : modélisation par deux courbes de Bézier indépendantes

Cet exercice et le suivant ont été préparés en cours/TD. Les notations utilisées sont celles du cours/TD.

Complétez la fonction `estimation_1` et le script `exercice_1.m`, afin de reproduire les résultats de la figure 1-d. Les inconnues sont les vecteurs $\beta^k = [\beta_1^k, \ldots, \beta_d^k]^\top$ et $\gamma^k = [\gamma_1^k, \ldots, \gamma_d^k]^\top$. Pour la $k^{\text{ème}}$ silhouette, les deux systèmes linéaires à résoudre s'écrivent sous la forme $A^k \beta^k = D^k$ et $A^k \gamma^k = E^k$ (les matrices de ces deux systèmes sont identiques). En posant $\delta^k = [\beta_1^k, \ldots, \beta_d^k, \gamma_1^k, \ldots, \gamma_d^k]^\top$, ces deux systèmes peuvent être réécrits sous la forme d'un seul système $F^k \delta^k = G^k$. La matrice F^k se déduit facilement de A^k, et le vecteur G^k est obtenu par concaténation des vecteurs D^k et E^k. Résolvez ce système au sens des moindres carrés ordinaires, en utilisant la *pseudo-inverse* (fonction `pinv` de Matlab) ou l'opérateur `\`.

Exercice 2 : modélisation par deux courbes de Bézier couplées

Comme cela a déjà été signalé, le sommet de la flamme n'est pas fermé, car les deux points de contrôle situés au sommet, à savoir $P_d^k = (\beta_d^k, 1)$ et $Q_d^k = (\gamma_d^k, 1)$, ne coïncident pas. Complétez la fonction `estimation_2` et le script `exercice_2.m`, de façon à coupler les deux courbes de Bézier associées à une même silhouette, en faisant en sorte que le point de contrôle situé au sommet de la flamme, c'est-à-dire à l'ordonnée $y = 1$, soit commun aux deux courbes. Il ne faut surtout pas moyenner les abscisses β_d^k et γ_d^k des deux bords, mais reformuler le problème sous la forme d'un nouveau système linéaire $\bar{F}^k \bar{\delta}^k = \bar{G}^k$, où $\bar{\delta}^k = [\beta_1^k, \ldots, \beta_{d-1}^k, \gamma_1^k, \ldots, \gamma_{d-1}^k, \epsilon^k]^\top$ et $\epsilon^k = \beta_d^k = \gamma_d^k$. Résolvez ce système au sens des moindres carrés ordinaires.

Exercice 3 : simulation de silhouettes par tirages aléatoires

L'analyse précédente permet de caractériser la silhouette d'une flamme par $2\,d-1$ paramètres. Pour simuler une silhouette, vous pourriez choisir aléatoirement des points de contrôle parmi ceux qui ont déjà été estimés, mais il est préférable de modéliser les distributions des abscisses des points de contrôle par des lois normales, puis d'utiliser les lois estimées pour procéder au tirage aléatoire de nouveaux points de contrôle (fonction `randn`).

Complétez le script `exercice_3.m` qui estime, pour chaque point de contrôle considéré comme une variable aléatoire, sa moyenne et son écart-type, à partir des réalisations empiriques que constituent les données réelles. Testez différentes valeurs du degré d, en n'oubliant pas de relancer le script `exercice_2.m` après chaque modification. Choisissez la valeur de d qui vous semble donner les silhouettes les plus « réalistes » (c'est souvent ainsi qu'on juge de la qualité d'un résultat en synthèse d'images).

Vous remarquez que, pour certaines silhouettes simulées, les bords gauche et droit se croisent (avec une fréquence qui dépend de la valeur de d). Modifiez le script `exercice_3.m` de telle sorte que les silhouettes présentant cette caractéristique ne soient pas affichées.

Exercice 4 : simulation de silhouettes texturées

Il est maintenant nécessaire de texturer les silhouettes obtenues à l'étape précédente. Si l'on souhaite que le résultat soit réaliste, il est recommandé de s'inspirer d'images réelles. L'image de la figure 1-e montre la texture moyenne normalisée, de taille 200×100, calculée à partir des n images réelles (vous avez déjà utilisé une partie de l'information contenue dans ces images). La difficulté du calcul de cette texture moyenne vient de ce que la silhouette diffère d'une image réelle à l'autre. C'est la raison pour laquelle chaque texture a été préalablement normalisée : chaque coupe de chaque flamme a été rééchantillonnée sur un même nombre de valeurs (égal à 100, en l'occurrence).

Complétez le script `exercice_4.m` afin de plaquer cette texture moyenne sur chacune des silhouettes obtenues par tirages aléatoires.