Extrapolation d'images binaires

Guillaume Charpiat

DEA MVA

29 mars 2002

Table des matières

Table des figures

1 Présentation du problème

On considère une image I, définie sur un domaine Ω, dont on ne connait les valeurs (i.e. les niveaux de gris) que sur $\Omega \backslash K$, où K est connexe. Comment extrapoler l'image sur K ?

Il ne s'agit pas ici de trouver une méthode d'extrapolation pratique ou efficace qui fonctionne quelle que soit l'image initiale I donnée parmis les configurations possibles, mais une méthode définie sur les images dites «naturelles», c'est-à-dire qui donne des résultats satisfaisants pour un œil humain sur des images qu'il juge «sensées».

Une façon de procéder consiste à essayer de relier les lignes de niveau de l'image d'un bout à l'autre du domaine K, et ce de manière «optimale». On se ramène ainsi à définir des critères de régularité sur des courbes.

Étant donnés deux points dans le plan munis chacun d'une tangente, l'on peut rechercher les courbes dont les extrémités coïncident à l'ordre 1 avec ces points, et qui minimisent la quantité

$$\int (\alpha \kappa^p(s) + 1) ds$$

où α est une constante positive arbitraire, $p \geqslant 1$ un réel arbitraire et $\kappa(s)$ la courbure à l'abscisse curviligne s.

Le cas particulier $p = 2$ est connu sous le nom de problème des «elastica».

2 Résolution théorique dans le cas $p = 2$

On se réfère pour ceci à l'article *Elastica and computer vision*, de D. Mumford. Certains points demandant quelques éclaircissements et justifications mathématiques, on refait la démonstration, comme suit.

Soit $\mathcal{C}$ une courbe dans $C^2([0; L], \mathbb{R}^2)$, dont les extrémités vérifient des conditions au bord à l'ordre 1, paramétrée par son abscisse curviligne $s \in [0; L]$.

Notons $\overrightarrow{t}(s) = \partial_s \overrightarrow{\mathcal{C}}(s)$ la tangente au point d'abscisse curviligne s, et $\overrightarrow{n}(s)$ la normale, définie comme étant l'image de $\overrightarrow{t}(s)$ par une rotation d'angle $+\frac{\pi}{2}$. On a ainsi $\partial_s \overrightarrow{t}(s) = \kappa(s) \overrightarrow{n}(s)$ où κ est la courbure (signée) de la courbe.

Minimiser $\int (\alpha \kappa^2(s) + 1) ds$ revient à minimiser $\int \kappa^2(s) ds$ sur l'ensemble des courbes de même longueur.

Considérons la courbe $\mathcal{D}$ définie par $\overrightarrow{\mathcal{D}}(s) = \overrightarrow{\mathcal{C}}(s) + \varepsilon \delta(s) \overrightarrow{n}(s)$, petite variation de $\overrightarrow{\mathcal{C}}$ qui préserve la longueur. La préservation de la longueur se traduit par :

$$\int_{s=0}^{L} \left\| \partial_s \overrightarrow{\mathcal{D}}(s) \right\| ds = L$$

2.1 Explicitation de la conservation de la longueur

On a, pour tout s dans $[0; L]$:

$$\begin{aligned}\partial_s \overrightarrow{D}(s) &= \overrightarrow{t}(s) + \varepsilon \partial_s \delta(s) \overrightarrow{n}(s) + \varepsilon \delta(s)(-\kappa(s) \overrightarrow{t}(s)) \\ &= [1 - \varepsilon \delta(s)\kappa(s)] \overrightarrow{t}(s) + \varepsilon \partial_s \delta(s) \overrightarrow{n}(s)\end{aligned}$$

$$\text{D'où} \quad \begin{aligned}\left\| \partial_s \overrightarrow{D}(s) \right\| &= \sqrt{(1 - \varepsilon\delta(s)\kappa(s))^2 + \varepsilon^2 \partial_s \delta(s)^2} \\ &= 1 - \varepsilon\delta(s)\kappa(s) + \mathrm{err}\left(\varepsilon\delta(s), \varepsilon\partial_s\delta(s)\right)\end{aligned}$$

où err : $\mathbb{R}^2 \to \mathbb{R}^+$ est telle que $\mathrm{err}(z) = o_{|z|\to 0}(z)$, c'est-à-dire $\lim\limits_{|z|\to 0} \dfrac{\mathrm{err}(z)}{|z|} = 0$. On a donc :

$$\begin{aligned}\int_{s=0}^{L} \left\| \partial_s \overrightarrow{D}(s) \right\| ds &= L - \int_{s=0}^{L} \varepsilon\delta(s)\kappa(s) ds + \int_{s=0}^{L} \mathrm{err}\left(\varepsilon\delta(s), \varepsilon\partial_s\delta(s)\right) \\ \int_{s=0}^{L} \delta(s)\kappa(s) ds &= \frac{1}{\varepsilon} \int_{s=0}^{L} \mathrm{err}\left(\varepsilon\delta(s), \varepsilon\partial_s\delta(s)\right)\end{aligned}$$

$$\text{Or } \forall s \in [0;L] : \quad \begin{aligned}\left|\mathrm{err}\left(\varepsilon\delta(s), \varepsilon\partial_s\delta(s)\right)\right| &\leqslant \sup_{s\in[0;L]} \left|\mathrm{err}\left(\varepsilon\delta(s), \varepsilon\partial_s\delta(s)\right)\right| \\ &\leqslant C_\varepsilon = \sup_{|z|\leqslant \varepsilon(\sup|\delta| + \sup|\partial_s\delta|)} |\mathrm{err}(z)|\end{aligned}$$

$$\text{Donc } \forall \varepsilon > 0 : \quad \left| \int_{s=0}^{L} \delta(s)\kappa(s) ds \right| \leqslant L \frac{C_\varepsilon}{\varepsilon}$$

Si l'on note $\beta(\varepsilon) = \operatorname*{Argsup}\limits_{|z|\leqslant \varepsilon(\sup|\delta| + \sup|\partial_s\delta|)} |\mathrm{err}(z)|$, on a $\beta(\varepsilon) \leqslant (\sup|\delta| + \sup|\partial_s\delta|)\varepsilon$ et $\lim\limits_{\varepsilon\to 0} \dfrac{C_\varepsilon}{\beta(\varepsilon)} = 0$ (propriété de err), d'où $\lim\limits_{\varepsilon\to 0} \dfrac{C_\varepsilon}{\varepsilon} = 0$.

Le critère de conservation de la longueur devient donc :

$$\int_{s=0}^{L} \delta(s)\kappa(s) ds = 0$$

2.2 Explicitation de la minimisation du critère de régularité

Exprimons maintenant que $\mathcal{C}$ est la courbe de longueur L qui minimise $\displaystyle\int_{s=0}^{L} \kappa^2(s) ds$.

Quelle que soit la fonction $\delta : [0;L] \to \mathbb{R}$ qui vérifie le critère de conservation de la longueur, la courbe correspondante $\mathcal{D} = \mathcal{C} + \varepsilon\delta\overrightarrow{n}$ doit vérifier que l'intégrale du carre de sa courbure est plus grande que $\displaystyle\int_{s=0}^{L} \kappa^2(s) ds$.

Notons $\sigma : [0;L] \to [0;L]$ la paramétrisation de D qui correspond à son abscisse curviligne, i.e. l'unique fonction strictement croissante qui vérifie :

$$\begin{aligned}\forall s \in [0;L] : \quad & \left\| \partial_s \overrightarrow{\mathcal{D}}(\sigma(s)) \right\| = 1 \\ \text{donc} \quad & \left\| \partial_s \sigma(s)\ \partial_\sigma \overrightarrow{\mathcal{D}}(\sigma) \right\| = 1 \\ & \partial_s\sigma(s) = \left. \frac{1}{\left\| \partial_\sigma \overrightarrow{\mathcal{D}}(\sigma) \right\|} \right|_{\sigma=\sigma(s)}\end{aligned}$$

Notons $\overrightarrow{t}_{\mathcal{D}}$ la tangente à la courbe $\mathcal{D}\circ\sigma$, $\overrightarrow{t}_{\mathcal{D}}$ la normale, et $\kappa_{\mathcal{D}}$ la courbure, paramétrés par l'abscisse curviligne de $\mathcal{D}\circ\sigma$. Par définition, on a :

$$
\begin{array}{llcl}
\forall s \in [0;L] : & \overrightarrow{t}_{\mathcal{D}}(s) & = & \partial_s \overrightarrow{\mathcal{D}}(\sigma(s)) \\
& & = & \partial_s \sigma(s)\ \partial_\sigma \overrightarrow{\mathcal{D}}(\sigma) \\
& & = & \frac{1}{\left\|\partial_\sigma \overrightarrow{\mathcal{D}}(\sigma)\right\|} \left([1-\varepsilon\delta\kappa]\ \overrightarrow{t} + \varepsilon \partial_\sigma \delta\ \overrightarrow{n}\right)\Big|_{\sigma=\sigma(s)} \\
& & = & \left.\dfrac{[1-\varepsilon\delta\kappa]\ \overrightarrow{t} + \varepsilon\partial_\sigma\delta\ \overrightarrow{n}}{\sqrt{(1-\varepsilon\delta\kappa)^2 + (\varepsilon\partial_\sigma\delta)^2}}\right|_{\sigma=\sigma(s)} \\
& & = & \overrightarrow{t} + \varepsilon\partial_\sigma\delta\ \overrightarrow{n}\big|_{\sigma=\sigma(s)} + O(\varepsilon^2)
\end{array}
$$

$$
\begin{array}{llcl}
\text{D'où} & \partial_s \overrightarrow{t}_{\mathcal{D}}(s) & = & \partial_s\sigma(s)\ \ \partial_\sigma \overrightarrow{t}_{\mathcal{D}}(\sigma)\big|_{\sigma=\sigma(s)} \\
& & = & \partial_s\sigma(s)\ \ \left((\kappa + \varepsilon\partial_\sigma^2\delta)\overrightarrow{n} - \varepsilon\kappa\partial_\sigma\delta\ \overrightarrow{t}\right)\big|_{\sigma=\sigma(s)} + O(\varepsilon^2) \\
& & = & (\kappa + \varepsilon(\partial_\sigma^2\delta + \delta\kappa^2))\overrightarrow{n} - \varepsilon\kappa\partial_\sigma\delta\ \overrightarrow{t}\big|_{\sigma=\sigma(s)} + O(\varepsilon^2)
\end{array}
$$

$$
\begin{array}{llcl}
\text{Or, comme} & \partial_s \overrightarrow{t}_{\mathcal{D}}(s) & = & \kappa_{\mathcal{D}}(s)\overrightarrow{n}_{\mathcal{D}}(s) \\
\text{On obtient :} & \kappa_{\mathcal{D}}^2(s) & = & (\kappa + \varepsilon(\partial_\sigma^2\delta + \delta\kappa^2))^2 + (\varepsilon\kappa\partial_\sigma\delta)^2|_{\sigma=\sigma(s)} + O(\varepsilon^2) \\
& & = & \kappa^2 + 2\varepsilon\kappa(\partial_\sigma^2\delta + \delta\kappa^2)|_{\sigma=\sigma(s)} + O(\varepsilon^2)
\end{array}
$$

$$
\begin{array}{llcl}
\text{D'où} & \displaystyle\int_{s=0}^{L} \kappa_{\mathcal{D}}^2(s)ds & = & \displaystyle\int_{s=0}^{L} \left(\kappa^2 + 2\varepsilon\kappa(\partial_\sigma^2\delta + \delta\kappa^2)\right)(\sigma(s))ds + O(\varepsilon^2) \\
& & = & \displaystyle\int_{s=0}^{L} \partial_s(\sigma(s))\left(\kappa^2 + \varepsilon\kappa(2\partial_\sigma^2\delta + \delta\kappa^2)\right)(\sigma(s))ds + O(\varepsilon^2) \\
& & = & \displaystyle\int_{s=0}^{L} \left(\kappa^2 + \varepsilon\kappa(2\partial_s^2\delta + \delta\kappa^2)\right)(s)ds + O(\varepsilon^2) \\
& & = & \displaystyle\int_{s=0}^{L} \kappa^2(s)ds + \varepsilon\int_{s=0}^{L} \delta(s)\left(2\partial_s^2\kappa(s) + \kappa^3(s)\right)ds + O(\varepsilon^2)
\end{array}
$$

$$
\begin{array}{lcl}
\text{D'où, pour tout } \varepsilon : \ \varepsilon\displaystyle\int_{s=0}^{L} \delta(s)\left(2\partial_s^2\kappa(s) + \kappa^3(s)\right)ds + O(\varepsilon^2) & > & 0 \\
\text{i.e.} \qquad\qquad\qquad \displaystyle\int_{s=0}^{L} \delta(s)\left(2\partial_s^2\kappa(s) + \kappa^3(s)\right)ds & = & 0
\end{array}
$$

On a donc : $\forall \delta \in C^2([0;L],\mathbb{R})$,

$$
\int_{s=0}^{L} \delta(s)\kappa(s)ds \ = \ 0 \implies \int_{s=0}^{L} \left(2\partial_s^2\kappa(s) + \kappa^3(s)\right)\delta(s)ds \ = \ 0
$$

Et donc $\exists A \in \mathbb{R},\ \ \forall s \in [0;L]$,

$$
2\partial_s^2\kappa(s) + \kappa^3(s) \ = \ A\kappa(s).
$$

2.3 Résolution de l'équation différentielle

En multipliant par $\partial_s\kappa$ et en intégrant, on obtient :

$$
(\partial_s\kappa)^2 + \frac{1}{4}\kappa^4 \ = \ \frac{A}{2}\kappa^2 + A'.
$$

2.3.1 Méthode des courbes elliptiques

En posant $u = \kappa$ et $v = 2\partial_s \kappa$, on a donc, pour deux constantes réelles B et C :

$$v^2 \;=\; -u^4 + Bu^2 + C$$

Ce qui nous donne :

$$(v + iu^2 + i\frac{B}{2})(v - iu^2 - i\frac{B}{2}) = C + \frac{B^2}{4}$$

Après de fastidieux calculs et rappels d'analyse complexe, on obtient pour solution la courbe suivante :

$$\overrightarrow{\mathcal{C}}(s) = c\,\partial_s \log F(s-\eta) - as$$

où

$$F(s) = \sum_{n \in \mathcal{Z}} e^{i\pi n^2 \frac{\omega_1}{\omega_2}} e^{2i\pi n \frac{s}{\omega_1}}$$

et $a, c, \eta, \omega_1, \omega_2$ sont à déterminer.

On ne détaillera pas ici cette méthode vu qu'elle est déjà décrite dans l'article de D. Mumford, qu'elle ne donne pas de résultat suffisamment explicite pour être directement exploitable (bien que la convergence de la série soit exponentielle) et qu'elle n'est pas adaptable à des puissances p quelconques. Le cas $p = 2$ constitue en effet un cas très particulier où l'on peut factoriser simplement $v^2 + u^4 - Bu^2$, point fondamental dans la suite du raisonnement.

2.3.2 Méthode simpliste explicite

On peut considérer le cas simple mais vraisemblable où l'annulation de la courbure implique l'annulation de sa dérivée. Pour justifier cette hypothèse supplémentaire, on peut remarquer que quand on parcourt la courbe et que la courbure s'annule, l'énergie ajoutée à chaque pas infinitésimal ds est minimale. L'annulation de la dérivée à ce moment précis permettra de rester plus longtemps à courbure nulle, et donc minimisera d'avantage l'énergie.

L'avantage est en fait principalement numérique. L'équation différentielle devient en effet :

$$(\partial_s \kappa)^2 + \frac{1}{4}\kappa^4 \;=\; \frac{A}{2}\kappa^2$$

Raisonnons tout d'abord «formellement», sans justifier nos calculs.

$$\begin{array}{lrcl}
\text{On a} & (\partial_s\kappa)^2 + \frac{1}{4}\kappa^4 & = & \frac{A}{2}\kappa^2 \\
\text{D'où} & \frac{\partial_s\kappa}{\sqrt{2A\kappa^2 - \kappa^4}} & = & \pm\frac{1}{2}
\end{array}$$

En notant f une primitive de $\frac{1}{\sqrt{2Ax^2-x^4}}$, on obtient :

$$\begin{array}{lrcl}
 & \partial_s f(\kappa(s)) & = & \pm\frac{1}{2} \\
\text{Donc} & \kappa(s) & = & f^{-1}\left(\frac{1}{2}(s-s_0)\right)
\end{array}$$

D'où l'intérêt de calculer explicitement f^{-1}.

$$\begin{aligned}
f(x) &= \int \frac{1}{x\sqrt{2A-x^2}}dx \\
&= \frac{1}{\sqrt{2A}}\left(\int \frac{1}{y\sqrt{1-y^2}}dy\right) \circ \left(\sqrt{2A}\,x\right) \\
&= \frac{1}{\sqrt{2A}}\left(\int \frac{1}{\sin z}dz\right) \circ \left(\arcsin(\sqrt{2A}\,x)\right) \\
&= \frac{1}{\sqrt{2A}} \ln\tan\left(\frac{1}{2}\arcsin(\sqrt{2A}\,x)\right)
\end{aligned}$$

$$\begin{array}{lrcl}
\text{D'où} & f^{-1}(x) & = & \frac{1}{\sqrt{2A}}\sin\left(2\arctan\left(e^{\sqrt{2A}\,x}\right)\right) \\
 & & = & \frac{1}{\sqrt{2A}\,\mathrm{ch}(\sqrt{2A}\,x)}
\end{array}$$

$$\begin{array}{lrcl}
\text{Et donc} & \kappa(s) & = & \frac{1}{\sqrt{2A}\,\mathrm{ch}\left(\sqrt{\frac{A}{2}}(s-s_0)\right)}
\end{array}$$

Et on vérifie que le κ ainsi obtenu est bien solution de l'équation différentielle.

Ceci est une solution générique dans les cas où la courbe ne possède pas de point d'inflexion. En pratique, on pourra l'utiliser pour interpoler un *snake* entre deux points consécutifs munis de tangentes orientées. En effet, si le *snake* comporte suffisamment de points, la portion litigieuse où figurera l'éventuel unique point d'inflexion sera assimilable à un segment.

3 Résolution théorique dans le cas p quelconque

3.1 Équation différentielle

On reprend les notations et la démarche du cas $p = 2$. Pour une longueur fixée L, on considère une courbe $\mathcal{C}$ qui minimise le critère :

$$\int_{[0;L]} \kappa^p(s)ds$$

où $\kappa(s)$ est la courbure de la courbe paramétrée par l'abscisse curviligne s. Pour toute variation C^2 δ telle que la courbe $\mathcal{D} = \mathcal{C} + \varepsilon\delta$ soit de longueur L à $O(\varepsilon^2)$ près, on a donc, d'après les calculs précédents,

$$\int_{s=0}^{L} \delta(s)\kappa(s)ds = 0$$

De même, la courbure $\kappa_{\mathcal{D}}(s)$ après reparamétrisation σ de $\mathcal{D}$ est donnée par :

$$\kappa_{\mathcal{D}}(s) = \kappa\left(1 + \varepsilon\left(\frac{\partial_\sigma^2\delta}{\kappa} + \delta\kappa\right)\right)\bigg|_{\sigma(s)} + O(\varepsilon^2)$$

Ce qui nous donne :

$$\begin{aligned}
\int \kappa_{\mathcal{D}}^p(s)ds &= \int \kappa^p\left(1 + p\varepsilon\left(\frac{\partial_\sigma^2\delta}{\kappa} + \delta\kappa\right)\right)(\sigma(s))\ ds \\
&= \int \kappa^p\left(1 + p\varepsilon\left(\frac{\partial_\sigma^2\delta}{\kappa} + \delta\kappa\right)\right)(1 - \varepsilon\delta\kappa)(\sigma(s))\ \ \partial_s\sigma(s)\ ds \\
&= \int \kappa^p\left(1 + \varepsilon\left(p\frac{\partial_s^2\delta}{\kappa} + (p-1)\delta\kappa\right)\right)(s)\ ds
\end{aligned}$$

Donc

$$\int p\kappa^{p-1}\partial_s^2\delta + (p-1)\delta\kappa^{p+1}(s)\ ds = 0$$

D'où :

$$p\ \partial_s^2\left(\kappa^{p-1}\right) + (p-1)\kappa^{p+1} = A\kappa$$

3.2 Résolution

En posant $f = \kappa^{p-1}$ et $\alpha = \frac{1}{p-1}$, on obtient donc :

$$\partial_s^2 f = Af^\alpha - \frac{1}{1+\alpha}f^{1+2\alpha}$$

D'où, pour une certaine constante C, et pour $\gamma = \frac{p}{p-1}$,

$$(\partial_s f)^2 = C + \frac{A}{\gamma}f^\gamma - \frac{1}{2\gamma^2}f^{2\gamma}$$

On aura donc $f(s) = F^{-1}(s - s_0)$ si F est une primitive de $\dfrac{1}{\sqrt{C + \frac{A}{\gamma}x^\gamma - \frac{1}{2\gamma^2}x^{2\gamma}}}$.

Malheureusement, ces primitives sont difficiles à expliciter. Néanmoins, on obtient des résultats comme précédemment pour le cas particulier $C = 0$, mais aussi pour le cas $\gamma = 1$, qui correspond à $p = \infty$.

Dans le cas $C = 0$, on obtient :

$$F(x) = \sqrt{\frac{\gamma}{A}}\ \frac{x^{1-\gamma/2}}{1-\gamma/2}\ \sum_{k=0}^{\infty}\frac{\Gamma(\frac{1}{2}+k)\ \Gamma(1/\gamma - \frac{1}{2}+k)\ \Gamma(\frac{1}{2}+\frac{1}{\gamma})}{\Gamma(\frac{1}{2})\ \Gamma(1/\gamma - \frac{1}{2})\ \Gamma(\frac{1}{2}+\frac{1}{\gamma}+k)\ \ k!}\left(\frac{x^\gamma}{2\gamma A}\right)^k$$

Il ne reste plus qu'à inverser cette fonction...

Dans le cas $\gamma = 1$, i.e. $p \to \infty$, on obtient :

$$F(x) = \sqrt{2} \arctan \left(\frac{x - A}{\sqrt{C + Ax - \frac{x^2}{2}}} \right)$$

D'où

$$F^{-1}(s) = g\left(\tan^2 \left(\frac{s}{\sqrt{2}} \right) \right)$$

où

$$g(y) = Ay \left(1 \pm \sqrt{1 + 2\frac{Cy - A^2}{A^2(y+2)}} \right)$$

Ce qui nous indique que le cas limite $\kappa = \lim\limits_{p \to \infty} f^{\frac{1}{p}}$ n'est pas régulier.

4 Schéma numérique (*snakes*)

On peut chercher une solution numérique par un algorithme de type itératif, en l'occurence, pour ce problème de minimisation, par une méthode utilisant des «snakes».

Soient A et B les deux extrémités (fixées) de la courbe. On appelle ici *snake* une suite $(P_i)_{1 \leqslant i \leqslant n}$ de points du plan, dont le premier et le dernier sont respectivement A et B, avec n entier quelconque.

On peut choisir comme terme à minimiser la quantité suivante :

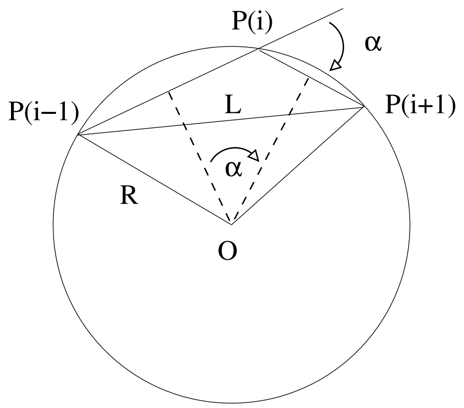
$$\frac{\alpha}{2(n-2)} \sum_{i=2}^{n-1} \left(\left\| \overrightarrow{P_{i-1}P_i} \right\| + \left\| \overrightarrow{P_i P_{i+1}} \right\| \right) \kappa^p(i) + \sum_{i=2}^{n} \left\| \overrightarrow{P_{i-1}P_i} \right\| + \beta \sum_{i=1}^{n/20} \left[d_A(P_i) + d_B(P_{n-i}) \right]$$

où $d_A(M)$ et $d_B(M)$ sont les distances respectives du point M aux tangentes en A et en B, et β un réel positif. Les deux premiers termes s'obtiennent par discrétisation directe de l'intégrale précédente, l'autre étant un terme d'attache aux données (tangentes aux extrémités).

Le premier problème préoccupant qui apparait est celui du calcul de la courbure qui, si trop local, pourrait s'avérer fantaisiste si le *snake* comporte des irrégularités. On peut donc chercher d'une part à lisser le *snake*, et d'autre part à exprimer la courbure en chaque point en tenant compte d'un voisinage arbitrairement grand. Nous allons vite nous rendre compte qu'il est intéressant de réaliser ces deux opérations simultanément.

4.1 Calcul de la courbure en s'appuyant sur trois points

Voici tout d'abord la méthode la plus simple pour calculer la courbure en un point P_i du *snake*. La courbure κ n'est autre que l'inverse du rayon du cercle osculateur au *snake* au point P_i, c'est-à-dire l'inverse du rayon du cercle qui passe par les trois points P_{i-1}, P_i et P_{i+1} (κ étant éventuellement nul, dans le cas où les trois points sont alignés).



Ceci nous donne :

$$\kappa = \frac{2\sin\alpha}{L}$$

qui s'exprime plus explicitement :

$$\kappa = 2\frac{\overrightarrow{P_{i-1}P_i} \wedge \overrightarrow{P_iP_{i+1}}}{\left\|\overrightarrow{P_{i-1}P_i}\right\| \left\|\overrightarrow{P_iP_{i+1}}\right\| \left\|\overrightarrow{P_{i-1}P_{i+1}}\right\|}$$

Le problème est que cette méthode suppose le *snake* lisse. Or, lorsque l'on va faire évoluer le *snake* pour effectuer la descente de gradient, des irrégularités risquent d'apparaitre.

4.2 Calcul de la courbure en s'appuyant sur le voisinage du point

On va donc chercher à calculer la courbure en s'appuyant sur un plus grand nombre de points. Considérons un ensemble de N points successifs P_i du *snake*. Cherchons le cercle, défini par son centre O et son rayon R, qui minimise la quantité :

$$H = \sum_i \left(\left\|\overrightarrow{OP_i}\right\|^2 - R^2 \right)^2$$

Au minimum, on a $\partial_R H = 0$ et $\overrightarrow{\nabla}_O H = \overrightarrow{0}$, ce qui nous donne :

$$\sum_i \left\|\overrightarrow{OP_i}\right\|^2 = NR^2$$

et

$$\sum_i \left(\left\|\overrightarrow{OP_i}\right\|^2 - R^2 \right) \overrightarrow{OP_i} = \overrightarrow{0}$$

En introduisant le point G barycentre des P_i, on obtient successivement :

$$\sum_i \left\|\overrightarrow{OP_i}\right\|^2 \overrightarrow{GP_i} = \overrightarrow{0}$$

$$\sum_i \left\|\overrightarrow{GP_i}\right\|^2 \overrightarrow{GP_i} + 2\sum_i \left\langle \overrightarrow{OG} \middle| \overrightarrow{GP_i} \right\rangle_2 \overrightarrow{GP_i} = \overrightarrow{0}$$

On obtient ainsi un système linéaire à deux équations et deux inconnues réelles $((\overrightarrow{OG})_x$ et $(\overrightarrow{OG})_y)$ dont la solution est donnée par :

$$\left\{\begin{array}{rcl}(\overrightarrow{OG})_x & = & \dfrac{1}{2}\dfrac{\sum_i \left\|\overrightarrow{GP_i}\right\|_2^2 \left[\langle g_x \,|g_y\rangle_N (\overrightarrow{GP_i})_y - \|g_y\|_N^2 (\overrightarrow{GP_i})_x\right]}{\|g_x\|_N^2 \|g_y\|_N^2 - \langle g_x \,|g_y\rangle_N^2} \\ \\ (\overrightarrow{OG})_y & = & \dfrac{1}{2}\dfrac{\sum_i \left\|\overrightarrow{GP_i}\right\|_2^2 \left[\langle g_x \,|g_y\rangle_N (\overrightarrow{GP_i})_x - \|g_x\|_N^2 (\overrightarrow{GP_i})_y\right]}{\|g_x\|_N^2 \|g_y\|_N^2 - \langle g_x \,|g_y\rangle_N^2}\end{array}\right.$$

en notant g_x et g_y les vecteurs à N coordonnées définis par $(g_x)_i = (\overrightarrow{GP_i})_x$ et $(g_y)_i = (\overrightarrow{GP_i})_y$, et en notant $\langle . \,|.\rangle_N$ et $\|.\|_N$ le produit scalaire usuel sur $\mathbb{R}^N$ et la norme associée.

L'expression du rayon de courbure devient :

$$R^2 = \left\|\overrightarrow{OG}\right\|_2^2 + \frac{1}{N}\left(\|g_x\|_N^2 + \|g_y\|_N^2\right)$$

On peut ainsi trouver le cercle qui approxime le mieux N points consécutifs P_i, et donc obtenir une approximation «lissée» de la courbure $\kappa = \frac{1}{R}$.

4.3 Lissage du *snake* par la courbure moyenne

Forts de ce résultat, essayons maintenant de l'utiliser pour lisser un *snake* comportant des irrégularités. Considérons le *snake* P formé des points $(P_i)_{1\leqslant i\leqslant n}$. Pour chaque point P_i on peut trouver grâce à la formule précédente le cercle osculateur des $|J_i|$ points $(P_j)_{j\in J_i}$ où $J_i = \{j; \min(1, i-m) \leqslant j \leqslant \max(n, i+m)\}$, m étant un entier arbitraire indiquant le nombre de points considérés pour construire le cercle. Notons O_i son centre et R_i son rayon. Notons également $H_i = \sum_{j\in J_i}\left(\left\|\overrightarrow{O_iP_j}\right\|^2 - R_i^2\right)^2$ et $H = \sum_i H_i$. On cherche à minimiser H, qui ne dépend que du *snake* P, les O_i et R_i étant des fonctions de P. Pour chaque point P_k, on est donc amené à calculer le gradient $\overrightarrow{\nabla}_{P_k}H$, qui vaut bien sûr $\sum_i \overrightarrow{\nabla}_{P_k}H_i$. Pour un k donné, en fixant tous les $P_l, l \neq k$, H_i peut être considéré comme une fonction de P_k seul.

$$\begin{array}{rcl}\overrightarrow{\nabla}_{P_k}H_i(P_k) & = & \overrightarrow{\nabla}_{P_k}H_i(O_i(P_k), R_i(P_k), P_k) \\ & = & \overrightarrow{\nabla}_{O_i}H_i(O_i, R_i, P_k)\overrightarrow{\overrightarrow{\nabla}}_{P_k}O_i(P_k) \;+\; \partial_{R_i}H_i(O_i, R_i, P_k)\overrightarrow{\nabla}_{P_k}R_i \\ & & \;+\; \overrightarrow{\nabla}_{P_k}H_i(O_i, R_i, P_k)\end{array}$$

Comme par définition O_i et R_i sont tels qu'à P_k fixé, ils minimisent $H_i(O_i, R_i, P_k)$, on a $\overrightarrow{\nabla}_{O_i} H_i = \overrightarrow{0}$ et $\partial_{R_i} H_i = 0$. Et donc :

$$\begin{aligned}
\overrightarrow{\nabla}_{P_k} H_i(P_k) &= \overrightarrow{\nabla}_{P_k} H_i(O_i, R_i, P_k) \\
&= \overrightarrow{\nabla}_{P_k} \sum_{j \in J_i} \left(\left\| \overrightarrow{O_i P_j} \right\|^2 - R_i^2 \right)^2 \\
&= 2 \sum_{j \in J_i} \left(\left\| \overrightarrow{O_i P_j} \right\|^2 - R_i^2 \right) \overrightarrow{\nabla}_{P_k} \left(\left\| \overrightarrow{O_i P_j} \right\|^2 - R_i^2 \right) \\
&= 4 \sum_{j \in J_i} \left(\left\| \overrightarrow{O_i P_j} \right\|^2 - R_i^2 \right) \left\| \overrightarrow{O_i P_j} \right\| \overrightarrow{\nabla}_{P_k} \left\| \overrightarrow{O_i P_j} \right\| \\
&= 4 \sum_{j \in J_i} \left(\left\| \overrightarrow{O_i P_j} \right\|^2 - R_i^2 \right) \left\| \overrightarrow{O_i P_j} \right\| \frac{\overrightarrow{O_i P_j}}{\left\| \overrightarrow{O_i P_j} \right\|} \delta_{j=k} \\
&= 4 \delta_{k \in J_i} \left(\left\| \overrightarrow{O_i P_k} \right\|^2 - R_i^2 \right) \overrightarrow{O_i P_k}
\end{aligned}$$

D'où :

$$\overrightarrow{\nabla}_{P_k} H = 4 \sum_{i; k \in J_i} \left(\left\| \overrightarrow{O_i P_k} \right\|^2 - R_i^2 \right) \overrightarrow{O_i P_k}$$

Pour lisser le *snake* au sens de H, il suffit d'ajouter à chaque P_k un vecteur $-\varepsilon \overrightarrow{\nabla}_{P_k} H$, où ε est positif.

4.4 Résultats numériques du lissage par la courbure

L'algorithme naturel qui découle des considérations précédentes, appliqué à un *snake* dont on fixe un certain nombre de points aux extrémités afin de préserver les tangentes, modifiera le *snake* de façon à ce que localement, il « ressemble » le plus possible à un cercle. Cependant, cela présente un inconvénient majeur : les points non fixés ont tendance à se regrouper par paquets de $N/3$ points, où N est le nombre de points consécutifs sur lesquels on s'appuie pour tracer les cercles osculateurs, de façon à minimiser l'écart entre les cercles et les points du *snake*.

Pour empêcher cela, on ajoute des «ressorts» entre les points du *snake*. À chaque instant t, à chaque couple de points consécutifs, on associe un ressort de longueur à vide L/n où L est la longueur de la courbe à l'instant $t-1$ et n le nombre de points qui la constituent. Ceci a pour conséquence immédiate d'homogénéiser la distribution des points le long de la courbe, et ce, sans changer ostensiblement la forme de la courbe.

Nous avons maintenant plusieurs paramètres à fixer : la proportion des points du *snake* qui sont fixés, le nombre de points que contient le *snake*, le rapport entre la force exercée par les ressorts et celle énoncée au paragraphe précédent. De fait, le choix de fixer un dixième des points s'avère satisfaisant, et le nombre de points que comporte le *snake* n'intervient pas qualitativement. On constate également que sur une assez large plage de valeurs du rapport entre les deux forces, les différences entre les différents schémas correspondants sont négligeables. Ce qui s'avère être qualitativement primordial, c'est la taille du voisinage considéré pour mesurer la «circularité», autrement dit, le rapport entre le nombre N de points consécutifs sur lesquel on s'appuie pour trouver les cercles, et le nombre n de points que comporte la courbe. Pour des valeurs faibles de N/n, le traitement est local, et on se contente de gommer les aberrations, tout en conservant la forme générale de la courbe. Pour des valeurs de l'ordre de $2/5$, ce lissage s'effectue en quelques pas de temps seulement, puis l'allure générale de la courbe se modifie, jusqu'à obtenir une courbe visuellement satisfaisante.

L'algorithme est assez robuste pour que l'on puisse ajouter au *snake* initial un bruit d'amplitude quelconque. En quelques pas de temps, les irrégularités disparaissent. On peut donc l'utiliser pour du débruitage de courbe.

Si pour une raison quelconque on ne veut appliquer l'algorithme qu'à des *snake* ne comportant qu'un petit nombre n de points, il peut être intéressant d'avoir un moyen d'extrapoler la courbe entre deux points consécutifs. Il y a plusieurs façons de s'y prendre. On peut utiliser la solution théorique générique que l'on a trouvée, puisque cette solution est valable dans les cas où il n'y a pas de point d'inflexion, et que, dans le pire des cas, il y aura une unique portion du *snake* qui en contiendra un. Le cas échéant, on extrapolera cette portion par un segment, la courbure y étant approximativement nulle. Pour appliquer cette méthode, il suffit donc de connaitre les tangentes à la courbe en chacun des points du *snake*. Ce qui n'est pas un problème, puisque l'on dispose des cercles osculateurs à la courbe. Mais ceci nous amène à considérer une seconde méthode : puisque l'on dispose des cercles osculateurs, autant interpoler en les représentant directement. Ceci conduit à une méthode générique d'extrapolation de *snakes* quelconques, consistant en l'application de la partie de l'algorithme qui fournit les cercles.

Les figures 1, 2 et 3 fournissent quelques illustrations du fonctionnement de l'algorithme.

L'ordre de grandeur de vitesse, pour un *snake* de 100 points, est de 200 pas de temps par seconde. Le débruitage s'avère être particulièrement efficace et rapide (de l'ordre de la dizaine de pas de temps) ; la déformation des courbes jusqu'à satisfaction visuelle est en revanche un peu plus lente (quelques centaines de pas de temps). Quant à l'interpolation, elle est effectuée en un unique pas de temps.

4.5 Déformation de *snake* par méthode variationnelle

Testons numériquement les critères de minimisation. Cela nous permettra par exemple d'obtenir les courbes solutions là où la théorie ne fournit pas de résultats explicites. L'implémentation est facilitée par le calcul de courbure «moyenne» énoncé au paragraphe précédent, qui est robuste. En pratique, on déforme un *snake*, comportant n points, dont on a fixé un certain nombre de points à chacune des extrémités, afin de préserver les tangentes (à la place de quoi on aurait pu ajouter un potentiel sur ces points notant l'écart aux tangentes). Le *snake* initial est formé de trois segments (les deux extrémités, plus le corps). Comme précédemment, on ajoute des ressorts entre les points du *snake* afin d'homogénéiser leur distribution le long de la courbe. La convergence du *snake* est assez lente : sur les exemples ci-après, il a fallu environ 3000 pas de temps pour atteindre l'état final, au rythme de 80 pas de temps par seconde.

Revenons au critère de minimisation $\int_{[0;L]} (\alpha\kappa^p + 1)\, ds$, et faisons varier p, ainsi

FIG. 1: Débruitage

Snake initial
(n = 100)

Snake bruité

... après 30
itérations

Snake bruité initial
(n = 100)

... après 10
itérations

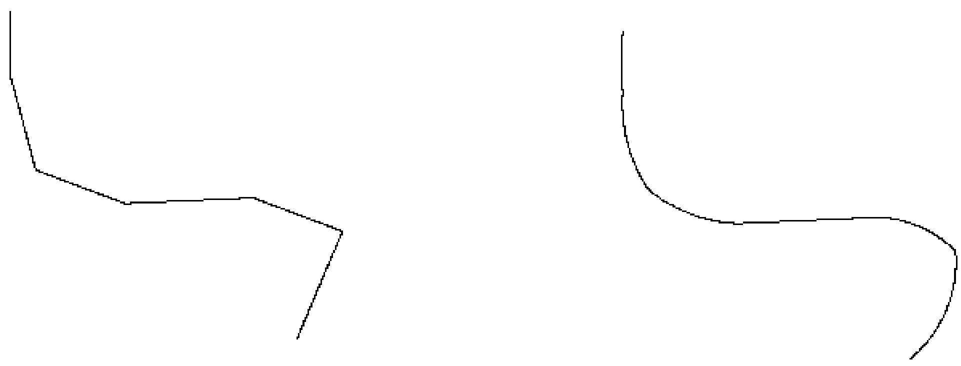
... après 50
itérations

Fig. 2: Déformation de courbes

Snake à l'état initial (n = 100) ... après 100 itérations

Snake à l'état initial (n = 100) ... après 1000 itérations

Fig. 3: Interpolation de courbes



Snake initial
(n = 7)

Snake interpolé

que les conditions initiales (position et angles aux extrémités). Désignons par «courbe en U» une courbe sans point d'inflexion ; «courbe en S» une courbe avec point d'inflexion, mais dont la variation totale d'angle est faible ; et «courbe en Z», une courbe avec point d'inflexion, avec variation importante d'angle, la courbure pouvant alors s'avérer importante (au niveau des extrémités).

La valeur de α ici est de l'ordre de 2000, pour des images de largeur type 256 pixels (voir la section suivante pour les remarques sur la dépendance de α avec la taille de l'image). Diminuer α revient à augmenter la tension de la courbe (comme lorsque l'on tend un élastique). Par exemple, lorsque l'on divise α par 10, les courbes s'apparentent pratiquement à des lignes brisées.

Il aurait été interessant de regarder ce qu'il se passe pour de grandes valeurs de p (afin d'avoir un aperçu du cas $p = \infty$), mais l'algorithme utilisé (descente de gradient classique) s'avère de plus en plus lent et de moins en moins fiable, les parties de la courbe de courbure faible ne jouant qu'un rôle très mineur dans le critère. Intuitivement, augmenter p revient à accorder un poids plus important aux parties de la courbe de courbure plus élevée, donc à minimiser la courbure maximale.

Il semblerait que les courbes en Z soient plus agréables visuellement pour de faibles valeurs de p (0.5, 1) : pour des valeurs plus grandes, les arcs de cercle aux extémités sont de courbure plus faible et par conséquent plus grands ; la jonction entre les deux se réalise de façon plus abrupte, devient verticale sur les exemples, moins «naturelle».

Pour les mêmes raisons, les courbes S et U sont plus agréables visuellement pour des valeurs de p plus grandes (2, 3) : une faible valeur de p favorisant l'apparition de portions de courbe de courbure élevée, on observe de brusques variations de la direction de la tangente à la courbe (la courbe en S se divise en trois segments ; le U devient carré : le bas de la courbe s'aplatit, pour p trop petit on peut diviser la courbe en trois morceaux, un horizontal, deux verticaux).

Les critères visuels semblent donc se reposer, pour les courbes en Z, sur l'abscisse relative des points d'ordonnée extrêmale, et, pour les autres, par la régularité de la courbure (donc une faible norme de la dérivée de la courbure).

Il apparait donc comme judicieux de tenir compte également dans les critères de minimisation de la dérivée de la courbure.

5 Extrapolation d'images binaires

5.1 Restriction à un domaine

5.1.1 Point de vue théorique

Lors de l'interpolation d'une courbe dans le cadre de la désocclusion d'images binaires, la courbe est contrainte à rester dans le domaine K de désocclusion. On peut voir cela comme l'ajout d'un potentiel V au critère de minimisation, valant 0 à l'intérieur de K et ∞ sur son complémentaire. Il convient de traiter le cas où K est un connexe fermé.

De fait, les solutions, minimisant localement le même critère que dans le cas sans contraintes, vérifient à l'intérieur de K les mêmes équations que précédemment, et

Fig. 4: Courbes solutions pour p = 0.5

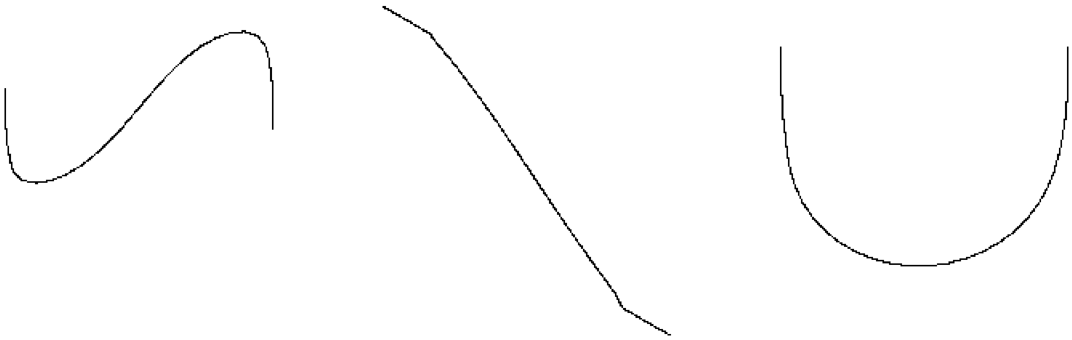


Fig. 5: Courbes solutions pour p = 1

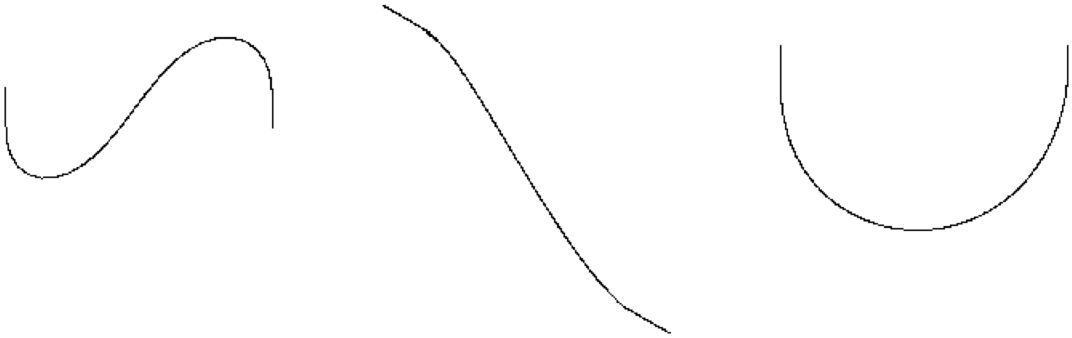
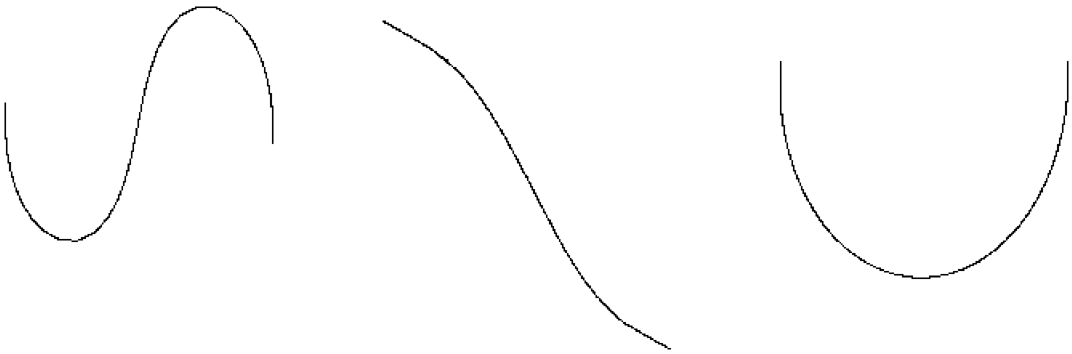


Fig. 6: Courbes solutions pour p = 2



Courbe en Z   Courbe en S   Courbe en U

Fig. 7: Courbes solutions pour p = 3

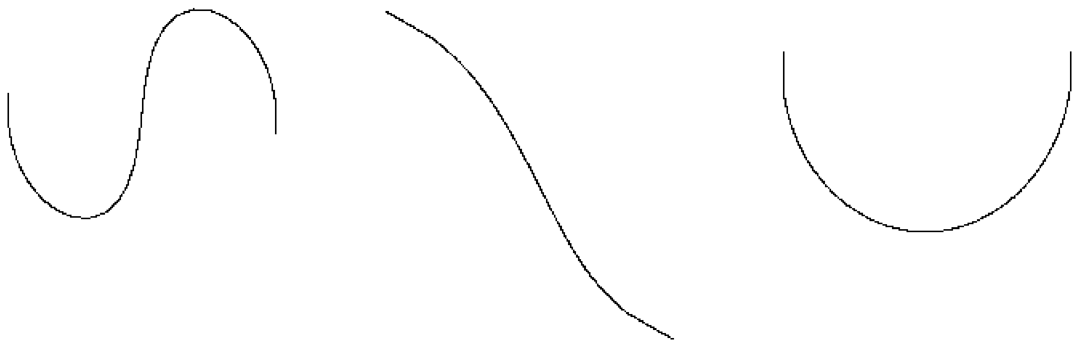


Fig. 8: Courbes solutions pour p = 4

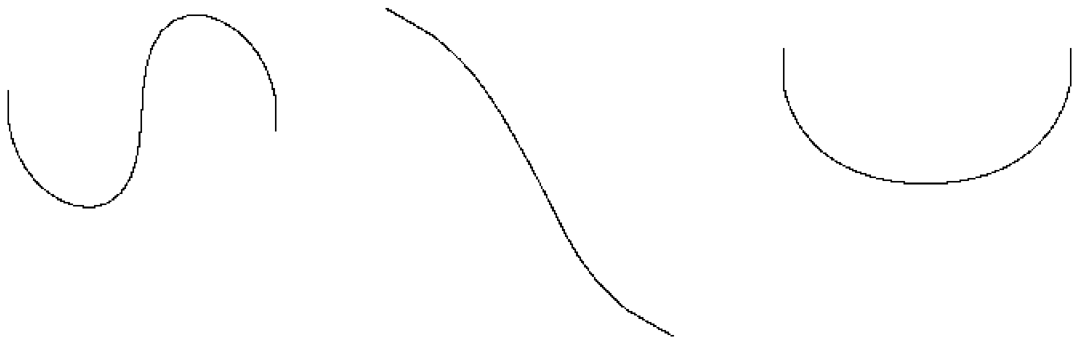
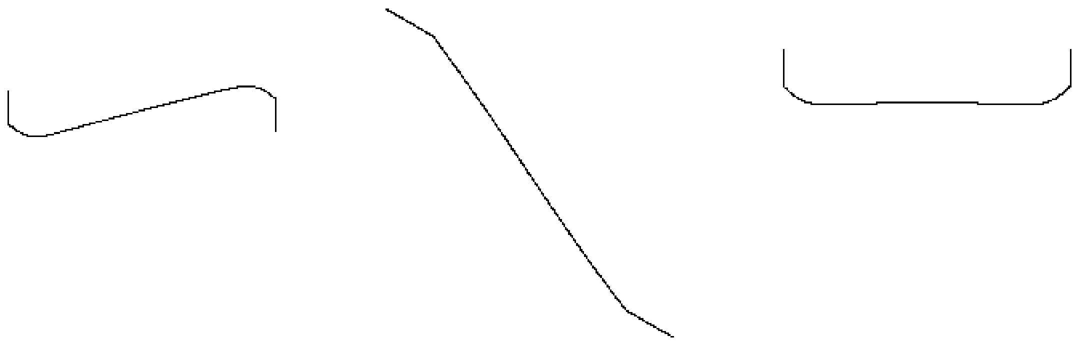


Fig. 9: Courbes solutions pour p = 16



Courbe en Z Courbe en S Courbe en U

font donc partie de la même famille de solutions ; on ne peut donc attendre d'expression explicite. Quant aux portions des solutions qui s'appuient sur le bord de K, elles sont justement données par le bord de K ; si celui-ci est suffisamment irrégulier, les solutions ne pourront pas s'appuyer contre, et feront donc partie intégrante de la famille de solutions au problème sans contrainte.

5.1.2 Implémentation

On peut choisir le potentiel V binaire énoncé précédemment si, dans l'algorithme, la méthode utilisée lors du calcul du gradient ramènera sans coup férir un point de la courbe un peu trop aventureux à l'intérieur de K ; ce dont on peut s'assurer en prenant pour conditions initiales un *snake* inclus dans K et en imposant qu'à chaque instant, la norme du déplacement de chacun des points du *snake* soit inférieure au pas utilisé pour calculer le gradient. En pratique, imposer $V = \infty$ également sur le bord de K conduira à une courbe strictement incluse dans K.

Une autre solution consiste à choisir un potentiel V nul à l'intérieur de K et de gradient suffisamment fort à l'extérieur. Par exemple, on peut considérer le potentiel distance à K : $V(P) = \alpha \; d(P, K)$, où α est une constante positive suffisamment grande. Le problème de cette seconde méthode est qu'il va falloir normaliser le déplacement du *snake*, puisque le potentiel met en jeu de grandes forces. Pour cette raison, on préfèrera appliquer directement la première méthode, qui demande moins de calculs.

On modélise le bord de K par un *snake*. Pour tester si un point P appartient à l'intérieur de K, on trace une demi-droite horizontale d'extrémité P et on compte le nombre de fois qu'elle coupe le *snake*. En pratique, le bord de K étant assimilable à un polygone (même si deux points consécutifs peuvent être distants d'une valeur subpixelique), on compte le nombre d'intersections des côtés du polygone avec la demi-droite.

Voici des exemples de courbes tracées sous contrainte, pour $p = 2$. Les courbes sont obtenues en environ 500 pas de temps, au rythme de 140 pas de temps par seconde. On remarque que la courbe est proche du bord par endroits, mais conserve sa régularité. On obtient des résultats similaires dans le cas où le bord est beaucoup plus anguleux. Cependant, de façon naturelle, on aurait peut-être plus attendu moins de contacts avec le bord pour la courbe Z, et plus de convexité pour la courbe U.

5.2 Combinatoire

5.2.1 Présentation du problème

Maintenant, attaquons-nous au vrai problème, qui consiste a relier, sous contrainte de rester dans un domaine connexe K, un certain nombre de points du bord de K entre eux, et ce, en minimisant un critère C.

Sur le bord de K figurent n points, notés $(A_i)_{1 \leqslant i \leqslant n}$, où n est un nombre pair. Ces points délimitent des zones homogènes du bord de K, alternativement noires et blanches.

On ne peut relier un point A_i qu'à un point d'indice de parités différentes, pour des raisons d'homogénéité des zones colorées. Pour chaque partition $\mathcal{P}$ de $[|1, n|]$ en

FIG. 10: Courbe Z sous contrainte

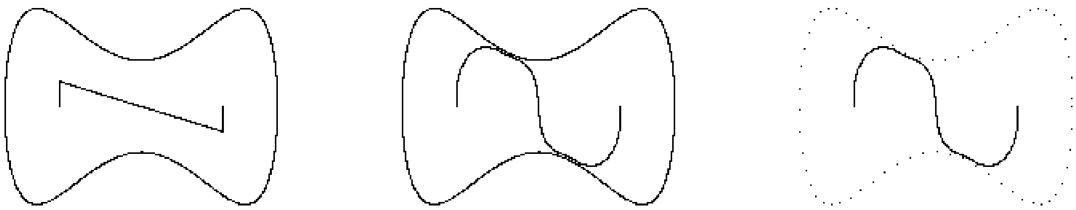


FIG. 11: Courbe S sous contrainte

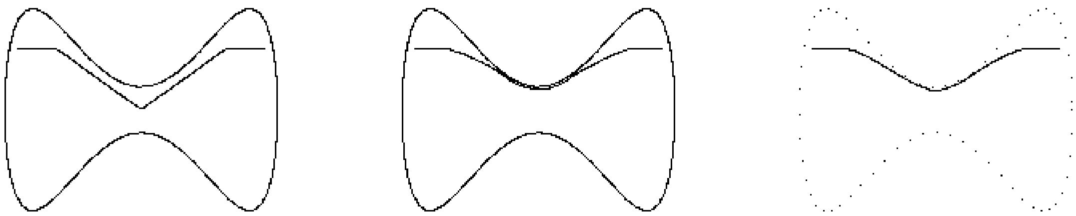
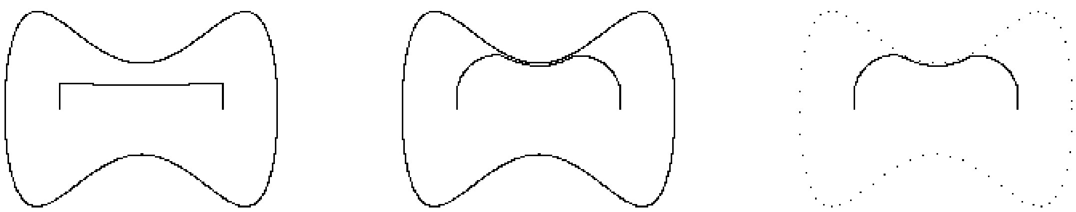


FIG. 12: Courbe U sous contrainte



Initialisation | Résultat | Résultat sans le bord paires $(P_k)_{1\leqslant k\leqslant \frac{n}{2}}$ d'éléments de parité différente, telle qu'il existe des courbes reliant chaque paire sans se couper, on cherche une configuration de courbes $\mathcal{C} = (\mathcal{C}_k)_{1\leqslant k\leqslant \frac{n}{2}}$ minimisant $E(\mathcal{C}) = \sum_k C(\mathcal{C}_k)$, chaque $\mathcal{C}_k$ reliant entre entre eux les deux éléments de la paire P_k.

La méthode de base consiste à calculer numériquement $E(\mathcal{P}) = \min_{\mathcal{C}} E(\mathcal{C})$ pour chaque partition $\mathcal{P}$, puis à retenir la configuration $\mathcal{C}$ qui minimise la plus petite des énergies $E(\mathcal{P})$.

L'un des problèmes posé par cette méthode est qu'il faut empêcher, pour une partition donnée, les courbes de se croiser, et que par la même occasion elles sont dépendantes les unes des autres, et donc qu'on ne peut pas les traiter séparément avec l'agorithme précédent. La solution retenue consiste à faire évoluer tour à tour chacune des courbes à chaque pas de temps, et de rajouter, lors du calcul des potentiels pour la déformation d'une courbe $\mathcal{C}_k$, un potentiel infini sur un voisinage de chacune des autres courbes. Si l'on souhaite se ramener à l'algorithme précédent, on peut utiliser l'astuce qui consiste à considérer le *snake* formé du bord de K et des deux courbes les plus proches de celle que l'on veut faire évoluer ; mais, en pratique, ceci ne permet pas d'imposer une distance relativement grande entre les différentes courbes.

Comment trouver l'ensemble des partitions pour lesquelles il existe des courbes reliant chaque paire sans se couper ? Ce problème ne dépend pas de l'ensemble K considéré, mais uniquement de l'ordre des A_i sur le bord de K. Pour le résoudre, on représente les partitions comme des arbres dont les noeuds sont des paires, et dont la relation de parenté est l'inclusion des indices ; par exemple $\{5, 6\}$ est le fils de $\{4, 7\}$. Les partitions qui peuvent se représenter sous forme d'un tel arbre sont celles que l'on cherche.

Le principal problème concerne l'initialisation des courbes. Il s'agit en effet de produire ex nihilo des courbes de contact d'ordre 1 avec des extrémités préfixées, qui ne sortent pas du domaine K, et qui ne se coupent pas entre elles. Dans le cas où K est convexe, il suffit de procéder comme dans la section précédente. Dans le cas non convexe, on pourrait penser à construire un difféomorphisme entre K et un disque, le bord de K ayant pour image le bord du disque. Les images des courbes cherchées seront alors les droites reliant les images des extrémités. Il est facile, à l'aide d'algorithmes itératifs, de faire converger un *snake* fermé vers un disque. Mais comment interpoler la bijection à l'intérieur de K ? Dans le cas où K est étoilé, on a une expression explicite de la bijection ; mais dans un cas quelconque ?

On reporte plutôt son attention sur une méthode de construction plus directe. On construit les courbes une à une, par récurrence, en commençant par celles qui doivent joindre deux points consécutifs. Pour une courbe donnée, on part d'une extrémité, et l'on suit le contour du *snake* à une certaine distance du bord, jusqu'à l'autre extrémité. Pour la courbe suivante, on considère le plus petit *snake* dont les bords sont inclus dans la réunion du précédent avec la dernière courbe construite, qui contient les extrémités à relier.

5.2.2 Résultats

Les contraintes exercées sur chacune des courbes apparaissent clairement. On peut d'ailleurs reprocher à l'algorithme, comme précédemment, de fournir des courbes qui peut-être accordent un trop grand poids aux contraintes, au détriment de la régularité. Ceci dit, les courbes obtenues sont tout de même convenables.

L'instant noté sur les figures comme «phase de croissance» correspond à environ 100 pas de temps ; pour l'équilibre entre les forces exercées entre les courbes, il faut attendre environ 1000 pas de temps, au rythme de 10 pas de temps par seconde. La rapidité de l'algorithme dépend très fortement de l'écart minimal imposé entre les courbes.

6 Autres façons d'aborder le problème

6.1 Pertinence du critère de minimisation

On cherche la puissance p de κ adéquate dans le critère de minimisation sur les courbes paramétrées par leur abscisse curviligne :

$$\int_{s=0}^{L} (\kappa^p + 1)\, ds$$

.

On a choisi la forme ci-dessus parce qu'elle prend en compte la longueur de la courbe et sa courbure. Il est naturel de chercher à minimiser la courbure, et de demander, parmis un ensemble de courbes vérifiant les mêmes critères sur leur courbure, de prendre celle de plus petite longueur. Mais ceci ne suffit pas à définir un critère. Il peut s'avérer opportun de s'imposer aussi des conditions d'échelle, comme : est-ce que si l'on multiplie par deux la distance entre les deux points fixant les extrémités de la courbe, on obtient la même courbe agrandie deux fois ? Pour ce faire : notons $\mathcal{C}$ une courbe minimisant le critère, L sa longueur et $\kappa(s)$ sa courbure. L'invariance par changement d'échelle α sur la courbe et les conditions initiales revient à dire que la courbe $\mathcal{D}$, définie par :

$$\overrightarrow{\mathcal{D}}(s) = \frac{1}{\alpha}\overrightarrow{\mathcal{C}}(\alpha s)$$

est solution du nouveau problème. Les caractéristiques de $\mathcal{D}$ sont :

La tangente :	$\partial_s \overrightarrow{\mathcal{D}}(s)$	$=$	$\overrightarrow{\mathcal{C}}'(\alpha s)$
D'où	$\partial_s^2 \overrightarrow{\mathcal{D}}(s)$	$=$	$\alpha\, \overrightarrow{\mathcal{C}}''(\alpha s)$
D'où la courbure :	$\kappa_{\mathcal{D}}(s)$	$=$	$\alpha\kappa(\alpha s)$
L'intégration :	ds	$=$	$d(\alpha s)/\alpha$
La longueur :	$L_{\mathcal{D}}$	$=$	$\frac{1}{\alpha}L$
Le critère de minimisation :	$\int_{s=0}^{L_{\mathcal{D}}} \kappa_{\mathcal{D}}^p(s) ds + L_{\mathcal{D}}$	$=$	$\alpha^{p-1}\int_{s=0}^{L} \kappa^p(s) ds + \frac{L}{\alpha}$

Donc le critère n'est pas invariant par changement d'échelle. Qualitativement, cela

Fig. 13: Cas convexe, avec 4 points à relier

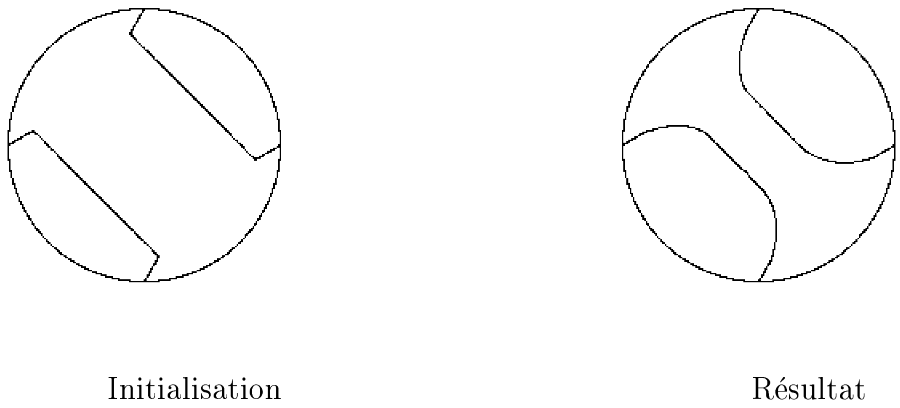
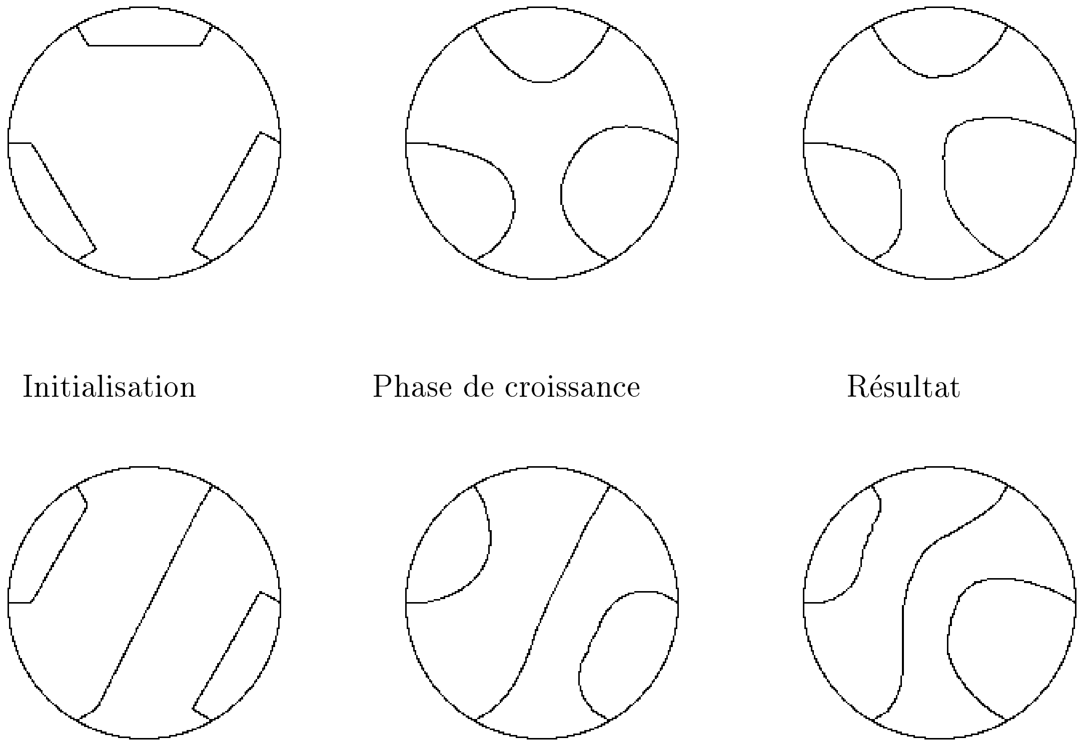


Fig. 14: Cas convexe, avec 6 points à relier, pour 2 partitions différentes



se traduit par le fait que $\mathcal{D}$ obtenu par zoom de $\mathcal{C}$ n'est pas la solution du problème avec zoom sur les conditions initiales.

Voici quelques invariants par zoom que l'on peut déduire des caractéristiques ci-dessus :

La courbure relative :	$L\kappa$	
Sa dérivée :	$L^2\partial_s\kappa$	
La variation d'angle :	$\int \kappa(s)ds$	
Un critère sur la courbure :	$\int \lvert\kappa(s)\rvert ds$	$L\int \lvert\partial_s\kappa(s)\rvert ds$
Un autre critère :	$L\int \kappa^2(s)ds$	$L^3\int(\partial_s\kappa(s))^2 ds$
Généralisation :	$L^{p-1}\int \kappa^p(s)ds$	$L^{2p-1}\int(\partial_s\kappa(s))^p ds$
Autrement dit :	$\frac{1}{L}\lVert L\kappa\rVert_p^p$	$\frac{1}{L}\lVert L^2\partial_s\kappa\rVert_p^p$
(moyenne de la courbure relative)		
Autre formulation :	$L^{\frac{p-1}{p}}\lVert\kappa\rVert_p$	$L^{\frac{2p-1}{p}}\lVert\partial_s\kappa\rVert_p$

On retiendra ces derniers critères, dont l'expression simple montre que leur minimisation, pour $p > 1$, reviendra à minimiser à la fois L et κ, $\partial_s\kappa$.

6.2 Méthode géodésique

On cherche un critère à minimiser, défini sur un ensemble de courbes dans le plan. Il s'agit donc tout d'abord de trouver des quantités intrinsèques aux courbes et pertinentes en regard du problème. On se ramène ainsi à ce qui a été fait dans le paragraphe précédent.

Mais, de fait, malgré l'équivalence courbe-courbure, c'est bien une courbe que l'on cherche, et les conditions aux bords sont difficilement exprimables en termes de courbure. Comme on demande aux extrémités un contact d'ordre 1, autant considérer les courbes $\mathcal{C} = (x, \alpha)$ dans $\mathbb{R}^2 \times \mathbb{S}_1$ qui vérifient :

$$\begin{array}{lrcl} \text{conditions de bord :} & \mathcal{C}(0) & = & (x_0, \alpha_0) \\ & \mathcal{C}(L) & = & (x_L, \alpha_L) \\ \text{condition de sens physique :} & \partial_s x(s) & = & e^{i\alpha(s)} \end{array}$$

et chercher parmis celles-ci celles qui minimisent un critère défini antérieurement. On se ramène donc, si le critère a une formulation adaptée, a la recherche de géodésiques dans un sous-ensemble de $\mathbb{R}^2 \times \mathbb{S}_1$, pour une certaine métrique (que l'on peut étendre à $\mathbb{R}^2 \times \mathbb{S}_1$ tout entier en complétant par ∞). Si l'on veut tenir compte de la dérivée de la courbure, en pratique, lors de l'implémentation, il peut s'avérer judicieux de considérer plutôt un sous-ensemble de courbes $\mathcal{C} = (x, \alpha, \kappa)$ de $\mathbb{R}^2 \times \mathbb{S}_1 \times \mathbb{R}$ où κ correspond à la courbure, sous-ensemble défini par les mêmes relations que précédemment, auxquelles il faut ajouter :

$$\partial_s\alpha(s) \;=\; \kappa(s)$$

Le principal problème est que tous les critères de minimisation énoncés plus haut ne se traduisent pas en termes de métrique locale, par exemple lorsqu'ils font intervenir la longueur de la courbe de façon non triviale. La solution ici considérée consiste à découper en deux le critère de minimisation de façon à séparer la longueur des termes portant sur la courbure. Par exemple, minimiser $L^{\frac{p-1}{p}}\lVert\kappa\rVert_p$ revient à minimiser :

$$(p-1)\ln\left(\int 1\right) + p\ln\left(\int \kappa^p\right)$$

Donc, pour appliquer une méthode numérique de recherche de géodésique, il suffit de connaitre à chaque pas de temps en tout point des chemins parcourus les valeurs de :

$$s = \int_{[0;s]} 1 \quad \text{et} \quad \int_{[0;s]} \kappa^p$$

le passage à l'instant suivant devenant alors naturel et explicite.

6.3 Critique de la démarche

Lors de l'interpolation de courbes se sont posés le choix du critère de régularité et le choix des conditions au bord. Mais, plus généralement, comment construire une courbe en en connaisant des morceaux ? Comment retirer l'information disponible dans ces fragments de courbe, et comment l'utiliser ?

On pourrait tout d'abord s'interroger sur l'ordre du contact qu'il faut garder aux extrémités de l'interpolation. Pourquoi ne pas choisir un ordre 2 plutôt que l'ordre 1, puisqu'il permet d'assurer la continuité de la courbure aux extrémités ? Quel critère choisir alors pour compléter la courbe ? En généralisant, on peut penser à compléter par des polynômes (ou splines) offrant le contact à l'ordre désiré, et même à compléter de façon analytique dans les cas qui s'y prêtent. Comme en pratique les images sont discrètes, on ne dispose pas assez d'information pour considérer la cas analytique. D'autre part, les polynômes offrent une piètre interpolation sur de grands intervalles.

En fait, si l'on dispose d'une courbe comportant des particularités flagrantes, comme périodicité de la courbure, ou présence répétée de motifs, ou bruit de statistiques connues, l'œil humain aimerait bien ne pas voir de «discontinuité dans la texture». On peut repérer ces «textures», ou motifs, par une décompostion de Fourier locale, c'est-à-dire par une décomposition en ondelettes. Cependant, comment reconstruire ensuite la partie manquante de la façon la plus vraisemblable possible ?

7 Conclusion

Dans le cadre de l'extrapolation d'images binaires, la méthode variationnelle appliquée au critère de minimisation $\int \kappa^2 + 1$ sur des *snakes*, aidé d'un calcul de courbure robuste, donne des résultats satisfaisants, et est adaptable à de nombreux cas (K non convexe, par exemple). Par ailleurs, il serait intéressant de développer une approche non pas basée sur juste des conditions de bords, mais également sur des caractéristiques ou des statistiques relevées sur les parties connues de la courbe à extrapoler.

Divers

Le langage de programmation utilisé est le C, les graphiques ont été faits à partir de la librairie graphique Xlib. Les calculs ont été menés dans la salle informatique (T) de l'ENS, sur des stations Sun Ultra-5 sous Solaris 8 (processeur sparc 400MHz).

Les figures ont été obtenues en traçant les courbes au pixel le plus proche, ce qui a pour inconvénient de faire apparaitre la discrétisation, mais a pour intérêt de mettre en évidence les qualités et les défauts des algorithmes proposés.

Référence

David Mumford, *Elastica and computer vision*

A Annexes

A.1 Équivalence courbe/courbure

Soit $\mathcal{C}$ une courbe de longueur L paramétrée par son abscisse curviligne s. Notons respectivement $\overrightarrow{\mathcal{C}}(s)$, $\overrightarrow{t}(s)$, $\overrightarrow{n}(s)$ et $\kappa(s)$ le point de la courbe d'abscisse curviligne s, la tangente, la normale ainsi que la courbure (signée) en ce point ; la normale étant définie comme l'image de la tangente par rotation de $\frac{\pi}{2}$. On a :

$$\partial_s \overrightarrow{\mathcal{C}}(s) = \overrightarrow{t}(s)$$

$$\partial_s^2 \overrightarrow{\mathcal{C}}(s) = \kappa \overrightarrow{n}(s)$$

avec

$$\forall s \in [0; L], \quad \left\| \overrightarrow{t}(s) \right\| = 1$$

Donc, si l'on utilise la notation complexe :

$$\partial_s^2 \overrightarrow{\mathcal{C}}(s) = i\kappa(s)\ \partial_s \overrightarrow{\mathcal{C}}(s)$$

Donc, quand on dispose de $\overrightarrow{\mathcal{C}}(s)$ et de ses dérivées, on obtient la courbure par :

$$\kappa(s) = -i \frac{\partial_s^2 \overrightarrow{\mathcal{C}}(s)}{\partial_s \overrightarrow{\mathcal{C}}(s)}$$

Réciproquement, quand on dispose de la courbure $\kappa(s)$, cela implique :

$$\partial_s \ln\left(\partial_s \overrightarrow{\mathcal{C}}(s)\right) = i\kappa(s)$$

D'où :

$$\overrightarrow{\mathcal{C}}(s) = \int_{u=0}^{s} e^{\int_{t=0}^{u} i\kappa(t)dt} du$$

à un déplacement près (c'est-à-dire à une translation et une rotation près). La donnée du point de départ $\overrightarrow{\mathcal{C}}(0)$ et de sa tangente $\partial_s \overrightarrow{\mathcal{C}}(0)$ fixe donc la courbe.

$$\overrightarrow{\mathcal{C}}(s) = \overrightarrow{\mathcal{C}}(0) + \partial_s \overrightarrow{\mathcal{C}}(0) \int_{u=0}^{s} e^{\int_{t=0}^{u} i\kappa(t)dt} du$$

A.2 Conditions aux bords

Les conditions portant sur le point initial ($\overrightarrow{\mathcal{C}}(0)$ et $\partial_s \overrightarrow{\mathcal{C}}(0)$) sont utilisées pour fixer le déplacement ci-dessus. Les conditions sur l'autre extrémité sont plus difficiles à expliciter, et surtout, à exploiter :

$$\overrightarrow{\mathcal{C}}(L) - \overrightarrow{\mathcal{C}}(0) = \partial_s \overrightarrow{\mathcal{C}}(0) \int_{u=0}^{L} e^{\int_{t=0}^{u} i\kappa(t)dt} du$$

$$\partial_s \overrightarrow{\mathcal{C}}(L) = \partial_s \overrightarrow{\mathcal{C}}(0) e^{\int_{t=0}^{L} i\kappa(t)dt}$$

Ce qui fixe les quantités :

$$\int_{t=0}^{L} \kappa(t)dt \quad \text{(variation d'angle)}$$

et

$$\int_{u=0}^{L} e^{\int_{t=0}^{u} i\kappa(t)dt} du$$

A.3 Réécriture du problème

Au lieu de chercher la courbe de longueur fixée L qui minimise le critère $\int_{[0;L]} \kappa^p(s)ds$, on peut chercher la fonction κ définie sur $[0; L]$ qui le minimise. κ est caractérisée par :

pour toute variation δ de κ qui vérifie :

$$\int_{t=0}^{L} \delta(t)dt = 0$$

et

$$\int_{u=0}^{L} \left(e^{\int_{t=0}^{u} i\kappa(t)dt} - 1\right) e^{\int_{t=0}^{u} i\delta(t)dt} du \ = 0$$

on a

$$\int_{[0;L]} (\kappa + \delta)^p \ \geqslant \ \int_{[0;L]} \kappa^p.$$

Le problème majeur qui se pose alors est la non-linéarité des conditions sur δ : si δ les vérifie, $\varepsilon\delta$ ne les vérifie pas, a priori. Pour simplifier les notations, on peut poser le problème en terme de primitives :

pour toute variation β de α qui vérifie :

$$\beta(0) = \beta(L)$$

et

$$\int_{u=0}^{L} \left(e^{i\alpha(u)} - 1\right) e^{i\beta(u)} du \ = 0$$

on a

$$\int_{[0;L]} (\partial_s\alpha + \partial_s\beta)^p \ \geqslant \ \int_{[0;L]} (\partial_s\alpha)^p\,.$$

Il s'agit maintenant de trouver des familles génériques de fonctions β qui vérifient les deux premiers critères. Avec un peu de calculs, on obtient une famille de fonctions β_{I_1,I_2} C^∞ par morceaux, nulles sur $[0; L]\backslash(I_1 \cup I_2)$ (avec I_1 et I_2 intervalles quelconques de $[0; L]$), qui s'expriment à l'aide d'arcsinus de termes en α. En faisant tendre $|I_1|$ et $|I_2|$ vers 0, et en exprimant le critère de minimisation, on peut faire apparaître des équations différentielles d'ordre 2 sur α. Mais il s'avère que la complexité des calculs incite à ne pas les achever, puisque, de toute façon, par la méthode présentée au début de ce rapport, l'équation différentielle est déjà connue.