



Approximation spline L1C1 par fenêtres glissantes pour le signal et l'image

Laurent Gajny, Éric Nyiri, Olivier Gibaru

► To cite this version:

Laurent Gajny, Éric Nyiri, Olivier Gibaru. Approximation spline L1C1 par fenêtres glissantes pour le signal et l'image. Journées du Groupe de Travail en Modélisation Géométrique 2013, Mar 2013, Marseille, France. hal-00841326

HAL Id: hal-00841326

<https://hal.archives-ouvertes.fr/hal-00841326>

Submitted on 4 Jul 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Approximation spline L_1C^1 par fenêtres glissantes pour le signal et l'image

Laurent Gajny¹, Éric Nyiri¹, Olivier Gibaru^{1,2}

¹ Arts et Métiers ParisTech, LSIS - UMR CNRS 7296, 8 Boulevard Louis XIV, 59046 Lille, France

² INRIA Lille-Nord-Europe, équipe NON-A, 40 avenue Halley 59650 Villeneuve d'Ascq, France

Résumé

Dans cet article, nous traitons le problème d'approximation de nuages de points par une courbe spline ou surface au sens de la norme L_1 . L'utilisation de cette norme permet de préserver la forme des données même en cas de changement brutal de celle-ci. Dans nos précédents travaux, nous avons introduit une méthode par fenêtre glissante de cinq points pour l'approximation courbe spline L_1 et une méthode de croix glissante de neuf points pour l'approximation surface spline L_1 de données type grille. Malgré leur complexité linéaire, ces méthodes peuvent demeurer lentes lorsqu'elles sont appliquées sur un large flot de données. Par conséquent, sur la base de nouveaux résultats algébriques sur l'approximation L_1 sur un nombre restreint de données, nous proposons ici des méthodes reposant sur des fenêtres de taille inférieure et nous comparons les différentes méthodes.

In this article, we address the problem of approximating scattered data points by C^1 -smooth polynomial spline curves and surfaces using L_1 -norm optimization. The use of this norm helps us to preserve the shape of the data even near to abrupt changes. In our previous work, we introduced a five-point sliding window process for L_1 spline curve approximation and a nine-point cross sliding window process for L_1 spline surface approximation of grid datasets. Nevertheless, these methods can be still time consuming despite their linear complexity. Consequently, based on new algebraic results obtained for L_1 approximation on restricted sets of points in both planar and spatial cases, we define in this article methods with smaller windows and we lead a comparison between the methods.

Mots-clés : Spline L_1 , approximation, méthode algébrique, préservation de forme, least absolute deviation, données bruitées.

de procédures telle qu'une intervention humaine sur le choix et la position des nœuds ou l'introduction de contraintes additionnelles.

1. Introduction

L'approximation de nuages de points est un problème important dans de nombreux domaines d'application. Dans ces problèmes de modélisation, une exigence classique est que les courbes ou surfaces générées possèdent des propriétés de conservation de forme. Ceci signifie que les courbes ou surfaces expriment les propriétés géométriques des données en accord avec la perception humaine. Lorsqu'on observe un changement brutal d'altitude ou d'espacement dans les données, les méthodes classiques d'approximation au sens des moindres carrés mènent à des oscillations (voir Figure 1). Ce phénomène peut être corrigé par une large gamme

Au cours des douze dernières années, les splines L_1 introduites par John E. Lavery et ses collaborateurs ont démontré une grande efficacité sur les problèmes d'optimisation globale d'interpolation et d'approximation (voir [CFL05, Lav00b, Lav00a, Lav04]). Ces méthodes corrigent le phénomène d'oscillation sans contraintes additionnelles (voir Figure 1). Dans [NGA11], les auteurs ont proposé le passage d'un traitement global à un traitement local des données en introduisant un processus de fenêtre glissante. En considérant des fenêtres de cinq points consécutifs pour l'interpolation par une courbe spline, ils ont proposé une méthode de complexité linéaire et rapide car algébrique avec de meilleures qualités de préservation de forme.

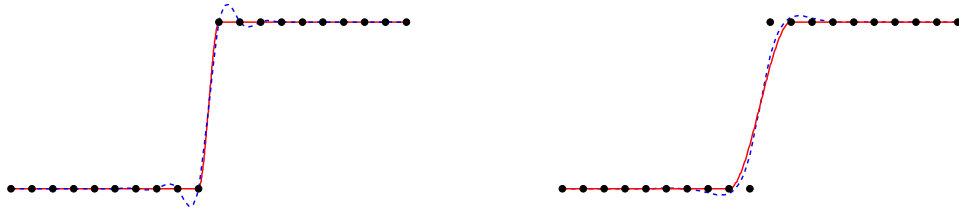


Figure 1: Interpolation (à gauche) et approximation (à droite) au sens L_2 (en pointillés bleus) et L_1 (ligne pleine rouge).

Le procédé de fenêtre glissante a été étendu dans [GNG13] au cadre de l'approximation spline pour un choix original de fonctionnelle de minimisation avec des contraintes de contrôle de distance. Le problème d'approximation L_1 sur cinq points est un problème très difficile qui n'est pas encore entièrement résolu algébriquement. C'est pourquoi, les méthodes basées sur ce type de fenêtre sont numériques et un algorithme type point intérieur est encore utilisé. Ceci implique un temps de calcul encore trop long dans le cas de large flot de données.

Dans cet article, sur la base de nouveaux résultats algébriques, nous investiguons des méthodes de fenêtres glissantes de taille différente et nous les comparons aux méthodes L_1 existantes.

2. Problème d'approximation L_1C^1 δ -contrôlé

Dans cette section, nous rappelons le problème d'approximation L_1 δ -contrôlé par une courbe spline de régularité C^1 introduit dans [GNG13] et nous mettons en avant quelques résultats théoriques d'intérêt.

Le théorème suivant définit notre problème d'optimisation \dagger et justifie sa cohérence.

Théorème 1 Soient $p_i = (x_i, y_i)$, $i = 1, 2, \dots, n$, n points dans \mathbb{R}^2 tels que $x_1 < x_2 < \dots < x_n$, une subdivision $\mathbf{u} = (u_i)_{i=1}^m$ telle que $\mathbf{u} \subseteq (x_i)_{i=1}^n$ et $\delta > 0$. Si $\mathcal{S}_{3,\mathbf{u}}$ représente l'espace vectoriel des splines cubiques C^1 de nœuds \mathbf{u} , alors :

$$\inf_{\gamma \in \mathcal{S}_{3,\mathbf{u}}} \int_{u_1}^{u_m} \left| \frac{d\gamma^2}{du^2}(u) \right| du \quad (1)$$

$$\text{tel que } |\gamma(x_i) - y_i| \leq \delta, \quad i = 1, 2, \dots, n, \quad (2)$$

existe et est atteint.

\dagger . Nous présentons ici une version planaire non-paramétrique pour plus de simplicité. Une version paramétrique dans \mathbb{R}^d est introduite dans [GNG13]

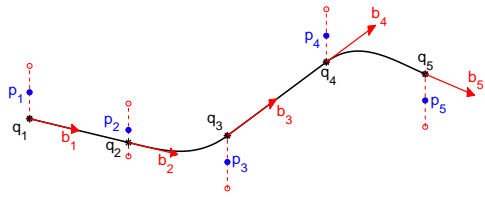


Figure 2: Les notations introduites pour le problème (3)-(4), les voisinages centrés aux points de données définis par (2) et une courbe spline solution (3)-(4).

Proof Voir [GNG13]. \square

Nous exigeons les contraintes (2) de manière à contrôler la déviation maximale aux points de données ce qui peut être utile, par exemple, pour les applications en planification de trajectoire. Ces contraintes définissent des voisinages des données illustrés en Figure 2. Étant donné que le placement des nœuds \mathbf{u} est un problème difficile en général et peut nécessiter une intervention humaine comme dans [Lav00a], nous considérons $m = n$. Ainsi, le problème (1)-(2) peut être aussi réécrit comme suit :

$$\min_{Q, B \in (\mathbb{R})^n} \Phi(Q, B) \quad (3)$$

$$\text{tel que } |q_i - p_i| \leq \delta, \quad i = 1, 2, \dots, n, \quad (4)$$

où

$$\Phi(Q, B) = \sum_{i=1}^{n-1} \int_{-\frac{1}{2}}^{\frac{1}{2}} |\Delta b_i + 6t(b_i + b_{i+1} - \frac{2}{\Delta u_i} \Delta q_i)| dt, \quad (5)$$

et $Q = \{q_i\}_{i=1, \dots, n}$ sont les points tels que $\gamma(u_i) = q_i$, $i = 1, \dots, n$ et $B = \{b_i\}_{i=1, \dots, n}$ sont les dérivées premières associées. Toutes ces notations sont illustrées en Figure 2.

La proposition suivante résout le problème (3)-(4) dans un cas particulier.

Proposition 1 Si il existe une droite passant par les voisinages des points de données p_i , $i = 1, 2, \dots, n$ définis par (4), alors les solutions du problème de minimisation (3)-(4) sont l'ensemble des telles droites.

Proof Notons que pour tout $Q, B \in \mathbb{R}^n$ définissant une droite, $\Phi(Q, B) = 0$. Donc, si une droite satisfait les contraintes (4), c'est une solution du problème (3)-(4). \square

Dans la proposition suivante, on résout algébriquement (3)-(4) pour $n = 3$.

Proposition 2 Soient $p_i = (x_i, y_i)$, $i = 1, 2, 3$ trois points de \mathbb{R}^2 , h_1 et h_2 les deux pentes associées et δ un réel strictement positif. On pose $\alpha = \frac{y_3 - y_1}{x_3 - x_1}(x_2 - x_1) + y_1$. Alors, les solutions de (3)-(4) pour $n = 3$ sont des droites si et seulement si :

$$y_2 \in [\alpha - 2\delta, \alpha + 2\delta]. \quad (6)$$

Si non, les solutions interpolent les points :

$$(x_1, y_1 + \text{sign}(h_1 - h_2)\delta), (x_2, y_2 - \text{sign}(h_1 - h_2)\delta), \\ (x_3, y_3 + \text{sign}(h_1 - h_2)\delta).$$

Remarque 1 La condition (6) signifie qu'il existe une droite intersectant les voisinages centrés en p_1 , p_2 et p_3 . Celle-ci est illustrée en Figure 5.

Nous avons besoin du lemme suivant, démontré dans [Auq07], afin de démontrer la proposition précédente.

Lemme 1 Pour q_1, q_2, q_3 trois réels fixés, on a :

$$\min_{b_1, b_2, b_3 \in \mathbb{R}} \sum_{i=1}^{m-1} \int_{-\frac{1}{2}}^{\frac{1}{2}} |\Delta b_i + 6t(b_i + b_{i+1} - \frac{2}{\Delta u_i} \Delta q_i)| dt \\ = 2 \left(\frac{\sqrt{10} - 1}{3} \right) \left| \frac{q_2 - q_1}{x_2 - x_1} - \frac{q_3 - q_2}{x_3 - x_2} \right|. \quad (7)$$

Proof[Proposition 2] On considère la position de p_3 par rapport à p_1 et p_2 . Les cas limites pour obtenir une solution droite sont que $(x_2, y_2 - \delta)$, respectivement $(x_2, y_2 + \delta)$, appartiennent aux droites interpolant $(x_1, y_1 - \delta)$ et $(x_2, y_2 + \delta)$, respectivement $(x_1, y_1 + \delta)$ et $(x_2, y_2 - \delta)$. Ceci mène aux conditions suivantes :

$$y_3 - \delta \leq \frac{y_2 - y_1 + 2\delta}{x_2 - x_1}(x_3 - x_1) + y_1 - \delta, \\ y_3 + \delta \leq \frac{y_2 - y_1 - 2\delta}{x_2 - x_1}(x_3 - x_1) + y_1 + \delta,$$

qui sont équivalentes à (6).

On suppose désormais que (6) n'est pas satisfaite. Philippe Auquiart a montré, dans [Auq07], le Lemme 1 permettant la minimisation par rapport aux dérivées premières b_i . Il reste

à minimiser par rapport aux points d'approximation q_i . Pour cela, on pose la fonction :

$$f : [y_1 - \delta, y_1 + \delta] \times [y_2 - \delta, y_2 + \delta] \times [y_3 - \delta, y_3 + \delta] \rightarrow \mathbb{R}, \\ (q_1, q_2, q_3) \mapsto \frac{q_2 - q_1}{x_2 - x_1} - \frac{q_3 - q_2}{x_3 - x_2}. \quad (8)$$

Cette fonction change de signe si et seulement si (6) est satisfaite. Sinon, on a :

$$\text{sign}(h_1 - h_2) = \text{sign}(f).$$

Compte-tenu de ce résultat, on peut résoudre le problème :

$$\min_{q_i, i=1,2,3} \left| \frac{q_2 - q_1}{x_2 - x_1} - \frac{q_3 - q_2}{x_3 - x_2} \right| \\ \text{tel que } \forall i = 1, 2, 3, q_i \in [y_i - \delta, y_i + \delta]. \quad (9)$$

Si $h_1 < h_2$, alors $\left| \frac{q_2 - q_1}{x_2 - x_1} - \frac{q_3 - q_2}{x_3 - x_2} \right| = \frac{q_3 - q_2}{x_3 - x_2} - \frac{q_2 - q_1}{x_2 - x_1}$. Le minimum de (9) est donc obtenu pour $q_1 = y_1 - \delta$, $q_2 = y_2 + \delta$ et $q_3 = y_3 - \delta$. Le cas $h_1 > h_2$ est similaire. \square

3. Fenêtres glissantes pour l'approximation L_1C^1 par une courbe spline

Les méthodes globales d'approximation développées par John E. Lavery et ses collaborateurs ont permis de mettre en évidence l'avantage d'utiliser cette norme. Cependant, ces méthodes peuvent être lentes sur de grands flots de données. Pour réduire le temps de calcul, on peut utiliser un procédé de fenêtre glissante. Sur chaque fenêtre centré en un point de donnée, on résout un problème local d'approximation ne tenant compte que des plus proches voisins, le nombre de voisins étant à choisir. Cette procédure permet à la fois de linéariser la complexité algorithmique mais aussi de tendre vers un algorithme algébrique. Nous présentons dans cette section deux choix de fenêtres pour des données planaires. Nous rappelons que l'étude de ce cas est suffisant même pour des données dans \mathbb{R}^d . En effet, dans ce cas, les données sont paramétrisées et on effectue une minimisation de type planaire dans chaque direction.

3.1. Méthode numérique par fenêtre glissante de cinq points

L'algorithme par fenêtre glissante de cinq points est directement inspiré de [NGA11] où le problème de l'interpolation L_1 est ainsi traité. Cette méthode est présentée en détails dans [GNG13]. Le procédé est illustré en Figure 3. Sur chaque fenêtre, on résout le problème d'approximation et on ne conserve que les informations au point central, point d'approximation et dérivée première.

Dans le cas de l'interpolation, le problème d'optimisation sur cinq points est résolu algébriquement. Cette

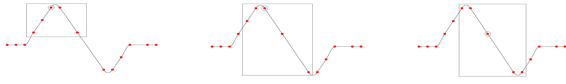


Figure 3: Fenêtre glissante de cinq points.

tâche s'avère beaucoup plus complexe pour l'approximation puisque les points de passage aux nœuds, les points d'approximation, sont maintenant des inconnues. Le problème d'approximation comporte donc deux fois plus d'inconnues que le problème d'interpolation. Une résolution numérique semble donc inévitable. Le Lagrangien du problème (10)-(11) n'étant pas différentiable partout, nous n'utilisons pas une méthode de multiplicateurs de Lagrange. Nous allons considérer le problème régularisé associé. La proposition suivante permet d'établir ce problème.

Proposition 3 Le problème (1)-(2) est équivalent à :

$$\min_{\gamma \in \mathcal{S}_{3,u}} \int_0^1 \left\| \frac{d\gamma^2}{du^2} \right\|_1 du \quad (10)$$

$$\text{tel que } \sum_{i=1}^n \|\gamma(t_i) - p_i - \delta\|_1 + \|\gamma(t_i) - p_i + \delta\|_1 = 4n\delta. \quad (11)$$

Proof Voir [GNG13]. \square

Le problème L_1/l_∞ (1)-(2) est donc équivalent à un problème L_1/l_1 . La démonstration de ce résultat présenté dans [GNG13] précise que la valeur $4n\delta$ est le minimum de la fonction de $x : \sum_{i=1}^n \|x - p_i - \delta\|_1 + \|x - p_i + \delta\|_1$. On peut donc déduire une version régularisée simple de (1)-(2) dont on vérifie la cohérence dans le résultat suivant.

Théorème 2 Sous les notations du Théorème 1 et avec λ un réel strictement positif, alors :

$$\inf_{\gamma \in \mathcal{S}_{3,u}} \int_0^1 \left\| \frac{d^2\gamma}{du^2}(u) \right\|_1 du + \lambda \sum_{i=1}^n \|\gamma(t_i) - p_i - \delta\|_1 + \|\gamma(t_i) - p_i + \delta\|_1 \quad (12)$$

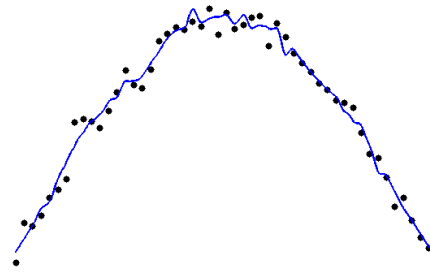
existe et est atteint.

Proof Voir [GNG13]. \square

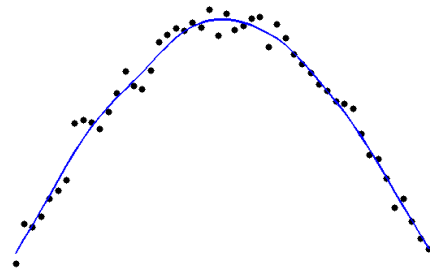
Cette version régularisée (12) vérifie le même résultat sur les solutions droites de (1)-(2) énoncé en Proposition 1. Néanmoins, il s'agit toujours là d'un problème non-linéaire. Nous discrétisons donc chaque intégrale de la somme, après les avoir préalablement découpé en $d \in \mathbb{N}^*$ intégrales, par une méthode de point milieu. Nous obtenons alors un problème matriciel à $(n-1)d + 2n$ lignes et $2n$ colonnes. Nous traitons ce problème sur toute fenêtres de cinq points consécutifs dans les données au moyen de l'algorithme

primal affine introduit dans [Van89].

Nous avons testé cette méthode d'approximation sur des données bruitées. Nous avons dégradé un sinus par un bruit gaussien de moyenne nulle et d'écart type 0.05. Nous illustrons ces résultats en Figure 4. Après une seule itération, en raison des contraintes de distance maximale aux données, la spline obtenue n'est pas lisse. Cependant, on peut itérer la méthode. À l'étape $k+1$, les points de données sont les points d'approximation calculés à l'étape k . L'utilisation de ce procédé démontre un phénomène de lissage tout en conservant la forme des données initiales.



Itération 1



Itération 4

Figure 4: Débruitage par itération successive de la méthode numérique à cinq points avec $\delta = 0.4$.

Afin de réduire le temps de calcul de cette méthode, on propose dans le paragraphe suivant une réduction de la taille de la fenêtre. Nous considérons un procédé de fenêtre glissante de trois points qui tire profit de la résolution algébrique sur trois points présentée en Proposition 2.

3.2. Méthode quasi-algébrique par fenêtre glissante de trois points

L'algorithme pour la méthode à trois points, basé sur la Proposition 2, est présenté ci-dessous.

Data : Données $\{p_i = (x_i, y_i)\}_{i=1, \dots, n}$, $\delta > 0$.

for $i = 2 \rightarrow n - 1$ **do**

$$\alpha = \frac{y_{i+1} - y_{i-1}}{x_{i+1} - x_{i-1}}(x_i - x_{i-1}) + y_{i-1};$$

if $\alpha - 2\delta \leq y_i \leq \alpha + 2\delta$ **then**

Étendre la fenêtre à p_{i-2} et/ou p_{i+2} ;

Déterminer une droite de régression L_1

contrôlée $y = ax + b$;

$$q_i^* = (x_i, ax_i + b);$$

else

$$h_{i-1} = \frac{y_i - y_{i-1}}{x_i - x_{i-1}};$$

$$h_i = \frac{y_{i+1} - y_i}{x_{i+1} - x_i};$$

$$q_i^* = (x_i, y_i - \text{sign}(h_{i-1} - h_i)\delta);$$

end

end

Interpoler les q_i^* par la méthode L_1C^1 (Voir [NGA11]).

Algorithme 1 : Approximation courbe spline L_1C^1 par fenêtre glissante de trois points.

L'algorithme 1 est composé de deux étapes majeures. La première étape calcule les points d'approximation q_i^* par regard de la Proposition 2. La deuxième étape calcule les dérivées première par interpolation L_1C^1 de ces points q_i^* . Nous choisissons cette approche car le faisceau de dérivées possible pour une résolution sur trois points est large en général. Il est difficile de choisir une valeur cohérente quand on ne considère que deux voisins directs. La méthode d'interpolation L_1C^1 étant algébrique, l'algorithme n'est pas pénalisé en temps de calcul.

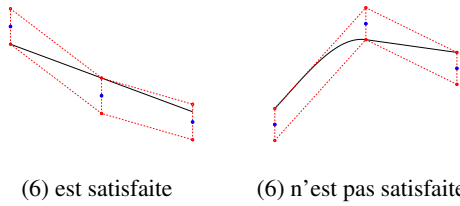


Figure 5: Les deux cas possibles dans l'algorithme à 3 points et une solution d'approximation L_1C^1

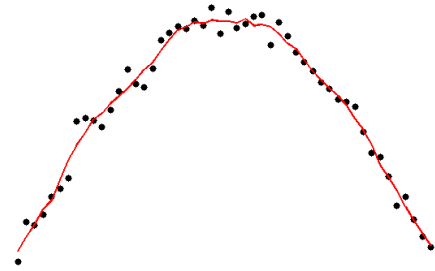
Les deux cas à considérer dans cette méthode sont illustrés en figure 5. Lorsque (6) n'est pas satisfaite, on obtient immédiatement le point d'approximation au centre de la fenêtre par application de la Proposition 2. Lorsque (6) est satisfaite, les solutions du problème sont un ensemble de droite qui est en général infini. Nous devons donc dégager dans cet ensemble l'une des plus cohérentes. Pour cela, on considère

des voisins supplémentaires. Nous passons alors d'une fenêtre de trois points à une fenêtre de cinq points et on introduit la méthode de régression linéaire L_1 contrôlée :

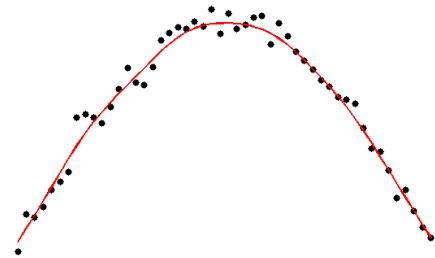
$$\min_{a, b \in \mathbb{R}} \sum_{i=1}^n \omega_i (|y_i - \delta - ax_i - b| + |y_i + \delta - ax_i - b|). \quad (13)$$

On s'intéresse au cas $n = 5$ et en particulier aux trois points centraux. On choisit donc $\omega_2 = \omega_3 = \omega_4 > \omega_1 = \omega_5 = 1$. Ainsi, s'il existe une droite intersectant les trois voisinages centraux, alors les solutions de (13) auront cette propriété. Ce problème peut être résolu par primal affine.

Nous avons testé, en Figure 6, cette méthode d'approximation sur le sinus bruité. Comme pour la méthode à cinq points, la courbe obtenue n'est pas suffisamment lisse après une seule itération. On itère donc la méthode selon le même principe pour des résultats similaires graphiquement (voir Figure 7).



Itération 1



Itération 4

Figure 6: Débruitage par itération successive de la méthode quasi-algébrique à trois points avec $\delta = 0.4$.

Nous avons effectué des expériences en dégradant plusieurs fois la même fonction de base $x \in [0, \pi] \mapsto \sin(x)$

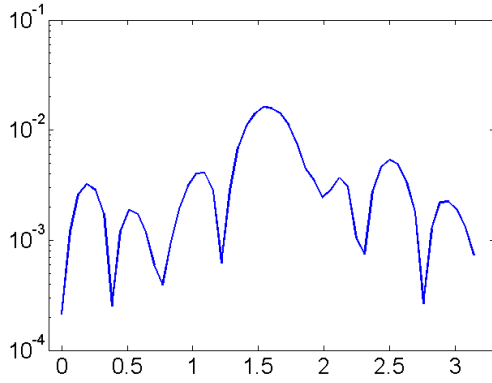


Figure 7: Écart dans un repère semi-logarithmique entre les solutions à trois et cinq points après quatre itérations des méthodes $\delta = 0.4$.

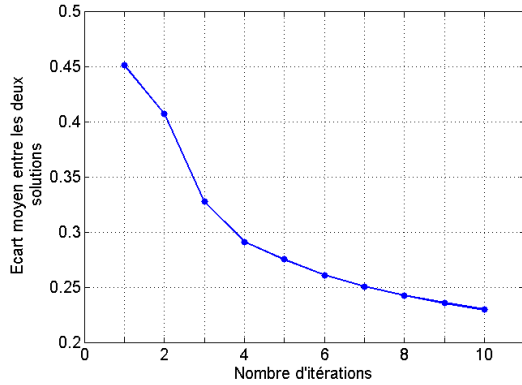


Figure 8: Évolution moyenne de l'écart, itération après itération, entre les deux méthodes.

par un bruit gaussien d'espérance nulle et d'écart type 0.05. Nous analysons l'écart moyen en norme 1 entre les solutions à trois et cinq points. Nous observons en Figure 8 une décroissance de cet écart itération après itération. De plus, la méthode à trois points, par son aspect quasi-algébrique, est plus rapide que la méthode à cinq points.

4. Fenêtres glissantes pour l'approximation L_1C^1 de données grille par une surface spline

Nous étendons les méthodes précédentes au cas des données organisées dans une grille régulière. Nous cherchons à approcher ce type de données par une surface de Bézier bicubique. Ce formalisme dans le cadre de l'optimisation L_1 peut s'avérer assez lourd numériquement (Voir [Zha07]). C'est pourquoi, nous ne cherchons pas directement une telle surface, mais les informations requises pour la construction de celle-ci uniquement à l'aide de courbes. Pour cela, on

utilise un processus de croix glissante comme illustré en Figure 9. Considérons une croix à $4n + 1$ points de centre $\mathcal{P} = (x_{\mathcal{P}}, y_{\mathcal{P}}, z_{\mathcal{P}})$, de composantes respectivement horizontales et verticales :

$$\mathcal{H} = \{\mathcal{P}_i = (x_i, y_{\mathcal{P}}, z_i), i = 1, 2, \dots, 2n\} \cup \mathcal{P},$$

$$\mathcal{V} = \{\mathcal{P}_i = (x_{\mathcal{P}}, y_i, z_i), i = 2n + 1, 2n + 2, \dots, 4n\} \cup \mathcal{P}.$$

Le problème d'approximation L_1 par un couple de splines cubiques sur une telle croix s'écrit :

$$\inf_{\gamma(\cdot, y_{\mathcal{P}}), \xi(x_{\mathcal{P}}, \cdot)} \int_{x_1}^{x_{2n}} \left| \frac{\partial^2 \gamma}{\partial u^2}(u, y_{\mathcal{P}}) \right| du + \int_{y_{2n+1}}^{y_{4n}} \left| \frac{\partial^2 \xi}{\partial v^2}(x_{\mathcal{P}}, v) \right| dv$$

tel que, pour tout $i = 1, 2, \dots, 4n$

$$|\gamma(x_i, y_{\mathcal{P}}) - z_i| \leq \delta, \quad \text{si } \mathcal{P}_i \in \mathcal{H},$$

$$|\xi(x_{\mathcal{P}}, y_i) - z_i| \leq \delta, \quad \text{si } \mathcal{P}_i \in \mathcal{V},$$

$$|\gamma(x_{\mathcal{P}}, y_{\mathcal{P}}) - z_{\mathcal{P}}| \leq \delta,$$

$$\gamma(x_{\mathcal{P}}, y_{\mathcal{P}}) = \xi(x_{\mathcal{P}}, y_{\mathcal{P}}),$$

(14)

avec δ un réel strictement positif. On peut montrer, en suivant la même démonstration que dans le cas courbe, que le minimum de (14) existe et est atteint.

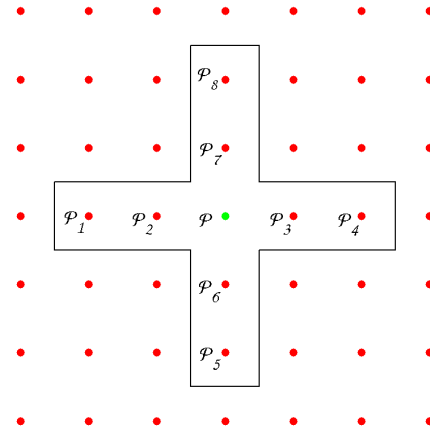


Figure 9: Croix de neuf points et notations pour l'approximation de données dans une grille régulière.

Nous allons étudier dans cette section deux méthodes d'approximation L_1 par croix glissantes, une méthode numérique à neuf points puis une méthode à cinq points avec résolution algébrique partielle. Ces méthodes sont appliquées au débruitage d'images.

4.1. Méthode numérique par croix glissante de neuf points

L'équivalent surfacique de la méthode courbe à cinq points est la méthode par croix glissante de neuf points illustrée en Figure 9. Nous effectuons une minimisation simulta-

née dans les deux directions de la grille. On cherche à minimiser sur les $Q_{k,l}$, $B_{k,l}$, $C_{k,l}$ - respectivement les points d'approximation, les vecteurs dérivées premières dans la direction de x et les vecteurs dérivées premières dans la direction de y - dans la croix centrée en $P_{i,j}$ la fonctionnelle suivante :

$$\begin{aligned} & \sum_{k=-2}^1 \int_{-\frac{1}{2}}^{\frac{1}{2}} \|\Delta^i B_{i+k,j} + 6t(B_{i+k+1,j} + B_{i+k,j} - 2\Delta^i Q_{i+k,j})\|_1 dt \\ & + \sum_{k=-2}^1 \int_{-\frac{1}{2}}^{\frac{1}{2}} \|\Delta^j C_{i,j+k} + 6t(C_{i,j+k+1} + C_{i,j+k} - 2\Delta^j Q_{i,j+k})\|_1 dt \\ & + \lambda \left(\sum_{k=-2}^2 (\|P_{i+k,j} - Q_{i+k,j} + \delta\|_1 + \|P_{i+k,j} - Q_{i+k,j} - \delta\|_1) \right. \\ & \left. + \sum_{k=-2}^2 (\|P_{i,j+k} - Q_{i,j+k} + \delta\|_1 + \|P_{i,j+k} - Q_{i,j+k} - \delta\|_1) \right), \end{aligned}$$

où Δ^i est l'opérateur de différence en i et Δ^j est l'opérateur de différence en j . Cette fonctionnelle n'étant pas linéaire, une discrétisation est nécessaire et on utilise l'algorithme primal affine pour la résolution. Comme dans le cas courbe, nous pouvons appliquer la méthode itérativement sur des données bruitées. Nous appliquons la méthode sur une image bruitée, avec un ratio signal à bruit de 25dB, en Figure 10.

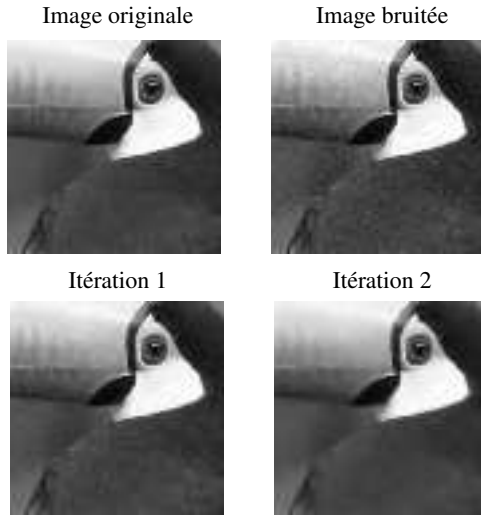


Figure 10: Expérience de débruitage avec la méthode de croix glissante à neuf points avec $\delta = 4$.

Nous obtenons de bons résultats graphiques. Nous conservons notamment les contours et nous lissons les parties uniformes de l'image. De plus, la complexité de la méthode est linéaire. Cependant, la résolution étant purement numérique, on ne peut garantir un résultat dans un temps raisonnable. Nous réduisons dans la section suivant la taille de la fenêtre et comparons les résultats pour les deux méthodes.

4.2. Méthode quasi-algébrique par croix glissante de cinq points

Comme dans le cas planaire, nous investiguons une fenêtre de taille moindre et on se tourne alors vers une croix glissante de cinq points. La proposition suivant est analogue à la Proposition 2 et sera la base de l'algorithme quasi-algébrique proposé dans ce paragraphe. En utilisant le Lemme 1 et les notations introduites en début de section, le problème (14) s'écrit dans le cas d'une croix de cinq points sous la forme suivante :

$$\begin{aligned} \min_{q_1, q_2, q_3, q_4} & \left| \frac{q - q_1}{x - x_1} - \frac{q_3 - q}{x_3 - x} \right| + \left| \frac{q - q_2}{y - y_2} - \frac{q_4 - q}{y_4 - y} \right| \\ \text{tel que} & \quad |q_i - z_i| \leq \delta, \quad i = 1, 3, \\ & \quad |q - z| \leq \delta, \end{aligned} \quad (15)$$

où les q_i sont les altitudes des points d'approximation recherchés. Pour ce type de croix, nous avons seulement deux configurations type possibles : le point cône et le point selle (Voir Figure 11).

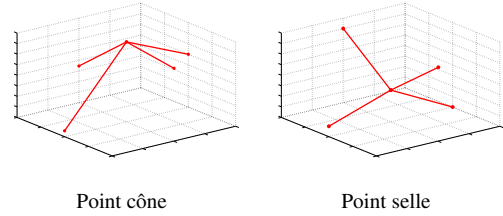


Figure 11: Les deux configurations possibles pour la méthode d'approximation L_1 à croix glissantes de cinq points.

La proposition suivante permet la résolution du problème (15) dans certains cas.

Proposition 4 Soit une croix à cinq points dans \mathbb{R}^3 de centre $\mathcal{P} = (x_{\mathcal{P}}, y_{\mathcal{P}}, z_{\mathcal{P}})$ et de composantes :

$$\mathcal{H} = \{P_i = (x_i, y_{\mathcal{P}}, z_i), i = 1, 2\} \cup \mathcal{P},$$

$$\mathcal{V} = \{P_i = (x_{\mathcal{P}}, y_i, z_i), i = 3, 4\} \cup \mathcal{P}.$$

Soient les pentes $h_1 = \frac{z_{\mathcal{P}} - z_1}{x_{\mathcal{P}} - x_1}$, $h_2 = \frac{z_2 - z_{\mathcal{P}}}{x_2 - x_{\mathcal{P}}}$, $h_3 = \frac{z_{\mathcal{P}} - z_3}{y_{\mathcal{P}} - y_3}$, $h_4 = \frac{z_4 - z_{\mathcal{P}}}{y_4 - y_{\mathcal{P}}}$ et δ un réel strictement positif. On pose :

$$\alpha = \frac{z_2 - z_1}{x_2 - x_1}(x_{\mathcal{P}} - x_1) + z_1 \quad \text{et} \quad \beta = \frac{z_4 - z_3}{y_4 - y_3}(y_{\mathcal{P}} - y_4) + z_4.$$

Si $z_{\mathcal{P}} \notin [\alpha - 2\delta, \alpha + 2\delta]$ et $z_{\mathcal{P}} \notin [\beta - 2\delta, \beta + 2\delta]$, alors les solutions du problème (15) satisfont la règle suivante :

- Si $h_1 < h_2$ et $h_3 < h_4$, alors $q_{\mathcal{P}} = z_{\mathcal{P}} + \delta$, $q_i = z_i - \delta$, $i = 1, 2, 3, 4$.
- Si $h_1 > h_2$ et $h_3 > h_4$, alors $q_{\mathcal{P}} = z_{\mathcal{P}} - \delta$, $q_i = z_i + \delta$, $i = 1, 2, 3, 4$.
- Si $h_1 < h_2$ et $h_3 > h_4$, alors $q_{\mathcal{P}} \in [z_{\mathcal{P}} - \delta, z_{\mathcal{P}} + \delta]$, $q_i = z_i - \delta$ pour $i = 1, 2$ et $q_i = z_i + \delta$ pour $i = 3, 4$.
- Si $h_1 > h_2$ et $h_3 < h_4$, alors $q_{\mathcal{P}} \in [z_{\mathcal{P}} - \delta, z_{\mathcal{P}} + \delta]$, $q_i = z_i + \delta$ pour $i = 1, 2$ et $q_i = z_i - \delta$ pour $i = 3, 4$.

Proof Ce résultat est un corollaire de la Proposition 2. On vérifie la compatibilité de la résolution du problème sur trois points dans les deux directions avec les conditions $z_P \notin [\alpha - 2\delta, \alpha + 2\delta]$ et $z_P \notin [\beta - 2\delta, \beta + 2\delta]$. \square

Dans le cas où $z_P \in [\alpha - 2\delta, \alpha + 2\delta]$ ou $z_P \in [\beta - 2\delta, \beta + 2\delta]$, nous résolvons le problème numériquement. Le problème matriciel est néanmoins de taille bien inférieure au problème sur cinq points. En effet, nous allons résoudre un problème approché en minimisant sur les q_i et q_P :

$$\left| \frac{q_P - q_1}{x_P - x_1} - \frac{q_2 - q_P}{x_2 - x_P} \right| + \left| \frac{q_P - q_3}{y_P - y_3} - \frac{q_4 - q_P}{y_4 - y_P} \right| + \lambda (|q_P - z_P + \delta| + |q_P - z_P - \delta|) + \lambda \left(\sum_{i=1}^4 |q_i - z_i + \delta| + |q_i - z_i - \delta| \right), \lambda > 0.$$

Ce problème, écrit sous forme matricielle, est de taille 12×5 sur chaque fenêtre alors que la matrice de la méthode à neuf points est de taille $\ddagger (8d + 19) \times 10$.

Cette méthode a été testée et comparée à la précédente sur une image bruitée (voir Figure 12). Au delà du gain en temps de calcul, nous obtenons de bons résultats graphiques et de meilleures valeurs de PSNR que la méthode précédente. On montre donc ici que seuls les plus proches voisins semblent avoir une importance réelle. Ceci conforte l'idée de l'utilisation de méthodes locales par processus de fenêtre glissante.

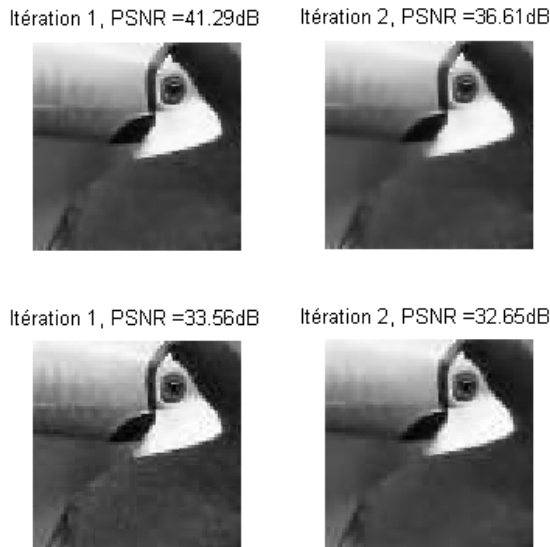


Figure 12: Expérience de débruitage avec la méthode à cinq points (haut) et neuf points (bas).

\ddagger . $1/d$ est le pas de discrétisation des intégrales.

5. Conclusion

Dans cet article, nous avons testé et comparé des méthodes d'approximation L_1 à fenêtre de tailles diverses. La méthode à trois points pour les courbes et la méthode à cinq points pour les surfaces tensorielles sont les plus rapides par leur aspect quasi-algébrique. De plus, elle démontre des résultats graphiques similaires aux méthodes à fenêtres plus larges. Une parallélisation des méthodes à petites fenêtres les rendra à l'avenir plus performantes en temps de calcul. L'utilisation de la méthode à trois points dans un contexte de robotique collaborative est à l'étude. Une amélioration graphique de la méthode surfacique sur les grilles serait des voisins "diagonaux". Ceci n'augmentera que très peu la taille du système matriciel. Cette méthode sera mise en place sous peu. Une extension possible de ce travail consistera au traitement de données désorganisées dans l'espace.

Références

- [Auc07] AUQUIERT P. : *Interpolation de points par des splines L^1 régulières*. PhD thesis, Université de Valenciennes et du Hainaut-Cambrésis, 2007.
- [CFL05] CHENG H., FANG S.-C., LAVERY J. E. : Shape-preserving properties of univariate cubic B-splines. *J. Comput. Appl. Math.* Vol. 174, Num. 2 (février 2005), 361–382.
- [GNG13] GAJNY L., NYIRI E., GIBARU O. : L_1C^1 polynomial spline approximation algorithms for large scattered data sets. *Soumis* (janvier 2013).
- [Lav00a] LAVERY J. E. : Shape-preserving, multiscale fitting of univariate data by cubic L_1 smoothing splines. *Computer Aided Geometric Design*. Vol. 17, Num. 7 (août 2000), 715–727.
- [Lav00b] LAVERY J. E. : Univariate cubic L_p splines and shape-preserving, multiscale interpolation by univariate cubic L_1 splines. *Computer Aided Geometric Design*. Vol. 17, Num. 4 (avril 2000), 319–336.
- [Lav04] LAVERY J. E. : Shape-preserving approximation of multiscale univariate data by cubic B-spline fits. *Comput. Aided Geom. Des.* Vol. 21, Num. 1 (janvier 2004), 43–64.
- [NGA11] NYIRI E., GIBARU O., AUQUIERT P. : Fast L_1C^k polynomial spline interpolation algorithm with shape-preserving properties. *Computer Aided Geometric Design*. Vol. 28, Num. 1 (janvier 2011), 65–74.
- [Van89] VANDERBEI R. J. : Affine-scaling for linear programs with free variables. *Math. Program.* Vol. 43, Num. 1 (janvier 1989), 31–44.
- [Zha07] ZHANG W. : *Bivariate cubic L_1 splines and applications*. PhD thesis, North Carolina State University, 2007.