

Ajustement automatique de surfaces paramétriques sur données de profondeur en présence d'un bruit hétérogène

F. Brunet^{1,2,3} A. Bartoli¹ N. Navab² R. Malgouyres³
¹ LASMEA (CNRS/UBP) ² CAMPAR (TU München) ³ LAIC (UdA)
Clermont-Ferrand (France) Munich (Allemagne) Clermont-Ferrand (France)
{Florent.Brunet,Adrien.Bartoli}@univ-bpclermont.fr
navab@im.tum.de
remy.malgouyres@laic.u-clermont1.fr

Résumé

Nous étudions le problème de l'ajustement de surfaces paramétriques continues sur des données de profondeur issues de caméras Temps-de-Vol (« Time-of-Flight cameras »). Une méthode d'ajustement sur de telles données doit être rapide (l'acquisition est en temps réel) et doit gérer un bruit hétérogène (le bruit dépend de la profondeur). Les méthodes classiques d'ajustement de surfaces ne répondent pas à ces critères.

Nous proposons une méthode d'ajustement de surfaces adaptée à de telles données. Notre méthode est rapide grâce à l'utilisation de B-splines et à l'exploitation de la disposition particulière des données considérées. Elle prend en compte l'hétérogénéité du bruit et préserve les discontinuités naturelles des scènes observées grâce à des ajustements locaux basés sur une segmentation des données en fonction de leur profondeur.

Mots clefs

Ajustement de surfaces, caméras Temps-de-Vol, B-splines, segmentation, régularisation

1 Introduction

Les caméras Temps-de-Vol (TdV) [1] acquièrent des données 2,5D en temps réel. Ces données se présentent sous la forme de cartes de profondeur discrètes c'est-à-dire d'images constituées de pixels dont la valeur représente la distance entre le capteur et le point observé [2]. Il est parfois intéressant, voire même nécessaire, de disposer d'une représentation analytique continue de la surface observée. Nous souhaitons ajuster une surface paramétrique comme un produit tensoriel de B-splines [3] sur des données TdV. Ajuster une surface sur de telles données représente un véritable défi.

Nous utilisons comme modèle de surface le produit tensoriel de B-splines. Afin de faire face aux problèmes liés au bruit entâchant les données, une contrainte de régularité doit être ajoutée pour lisser la surface ajustée [4, 5]. L'importance relative donnée à cette contrainte dépend de l'intensité du bruit : plus il est fort et plus la régularisation doit être importante. Dans les méthodes existantes, il est fait l'hypothèse que le bruit est de même nature et de même intensité pour l'ensemble des données. L'import-

tance de la régularisation est alors déterminée de manière globale. Or, avec les données que nous considérons, le bruit est *hétérogène* c'est-à-dire que son intensité n'est pas constante et dépend principalement de la profondeur [6]. Par conséquent, une approche globale n'est pas appropriée. Par exemple, si une partie des données est fortement bruitée, l'utilisation d'une approche globale entraînera une régularisation importante, ce qui n'est pas souhaitable pour les parties moins bruitées des données. De plus, une régularisation importante aura tendance à éliminer les discontinuités naturelles des scènes observées.

Nous proposons une méthode d'ajustement de surfaces permettant de résoudre les difficultés liées à la nature particulière des données TdV. La variabilité de l'intensité du bruit et les discontinuités des scènes sont prises en compte en effectuant une étape de segmentation des données en fonction des profondeurs. Un *ajustement local* est ensuite effectué sur chacune des parties segmentées. Afin d'obtenir une unique surface globale, les ajustements locaux sont regroupés en utilisant de manière judicieuse la propriété d'influence locale des points de contrôle des B-splines.

Les caméras TdV fournissent en temps réel des cartes de profondeurs constituées d'un grand nombre de pixels (de l'ordre de vingt mille). La méthode que nous proposons relève le défi d'être performante malgré cette quantité de données. Pour cela, nous exploitons la nature tensorielle des B-splines, la disposition sous forme de grille des données et les propriétés du produit de Kronecker.

Notre méthode adresse aussi le délicat problème que constitue la détermination de l'importance de la régularisation. À cet égard, nous proposons une adaptation du principe de divergence (« discrepancy principle ») et montrons la possibilité de son utilisation à partir de données TdV.

Organisation de l'article. Un état de l'art sur les méthodes globales d'ajustement de surfaces à partir de données de profondeur est donné à la section 2. Nous intégrons à cette section nos contributions concernant les problèmes liés à la régularisation. La section 3 explique comment notre méthode permet d'ajuster une surface paramétrique globale en fusionnant des ajustements locaux effectués sur des sous-ensembles de données qui sont obtenus par segmentation des données initiales en fonction des pro-

fondeurs. Enfin, la section 4 concerne les résultats expérimentaux.

Notations. Les valeurs scalaires sont écrites en caractères italiques (x), les vecteurs en caractères gras (\mathbf{p}) et les matrices en caractères sans serif (M). Le coefficient situé à l'intersection de la i -ème ligne et j -ème colonne d'une matrice A est noté $A_{i,j}$. La transposition est notée grâce au symbole T (\mathbf{p}^T, M^T). $[a, b]$ représente un intervalle continu fermé et $\llbracket c, d \rrbracket$ représente un intervalle discret ($\llbracket c, d \rrbracket = \{c, c+1, \dots, d\}$). Le produit cartésien de deux ensembles est noté avec le symbole \times . Le produit de Kronecker [3] entre deux matrices est représenté par le symbole \otimes . Les opérateurs de dilatation et d'érosion de la morphologie mathématique sont représentés respectivement par les symboles \oplus et \ominus . La fonction vec effectue la vectorisation colonne d'une matrice. La fonction sd calcule l'écart type de l'ensemble des coefficients d'une matrice ou d'un vecteur.

2 Ajustement global de surface

Les données que nous utilisons dans cet article sont des cartes de profondeur. Une telle carte est représentée par une matrice $Z \in \mathbb{R}^{m \times n}$. La valeur $Z_{i,j}$ est la profondeur du pixel (i, j) . Ce genre de données est parfois appelé *données 2,5D*. Une illustration¹ est présentée en figure 1.

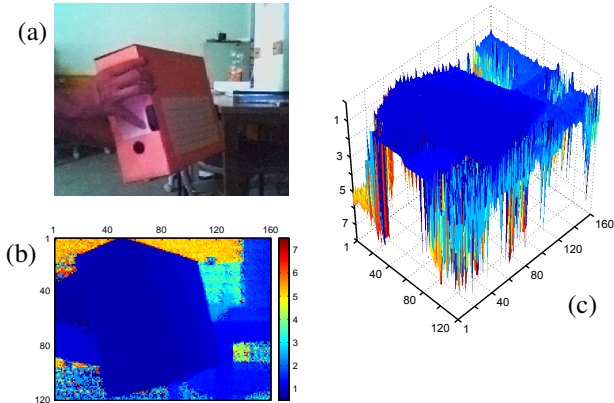


Figure 1 – (b) Exemple de carte de profondeur acquise avec une caméra PMD Tec 19k et (c) vue 3D de cette même carte. (a) Aperçu de la scène pris par un capteur CMOS. Les profondeurs sont exprimées en mètres.

Le problème de l'ajustement d'une surface sur une carte de profondeur consiste à déterminer une fonction $f : [1, n] \times [1, m] \rightarrow \mathbb{R}$ qui approxime les données c'est-à-dire telle que $f(j, i) \approx Z_{i,j}$. De plus, les fortes variations de f sont pénalisées afin d'obtenir une surface « lisse ».

L'approche paramétrique classique permettant d'ajuster une surface consiste à fixer un modèle puis à déterminer les paramètres de ce modèle de manière à ce que nos contraintes soient satisfaites. Dans cet article, nous utilisons comme modèle les produits tensoriels de splines mo-

nodimensionnelles définies sur la base des B-splines² [3]. Les paragraphes suivants expliquent en détail la méthode utilisée pour ajuster une surface de type B-spline sur des données de profondeur disposées sur une grille. Lorsque les données sont l'ensemble des profondeurs initiales Z , nous appelons cette méthode la *méthode globale*. De plus, nous verrons comment exploiter la nature tensorielle du modèle B-spline, la disposition particulière des données et les propriétés du produit de Kronecker pour obtenir une méthode rapide d'ajustement de surface. En particulier, nous proposons une méthode de régularisation s'intégrant dans ce schéma. Enfin nous verrons comment sélectionner automatiquement l'importance de la régularisation.

B-splines bidimensionnelles. Étant données deux suites croissantes $L = \{\lambda_{-k} \leq \dots \leq \lambda_{g+k+1}\}$ et $M = \{\mu_{-l} \leq \dots \leq \mu_{h+l+1}\}$ appelées *vecteurs nodaux*, une B-spline est une fonction s définie sur $[\lambda_{-k}, \lambda_{g+k+1}] \times [\mu_{-l}, \mu_{h+l+1}]$ par :

$$s(x, y) = \sum_{i=-k}^g \sum_{j=-l}^h P_{j+l+1, i+k+1} N_{i, k+1}(x) N_{j, l+1}(y)$$

Les entiers k et l sont les degrés selon la direction des abscisses et des ordonnées respectivement. La matrice $P \in \mathbb{R}^{(h+l+1) \times (g+k+1)}$ est la matrice des points de contrôle de la B-spline. Les entiers g et h déterminent le nombre de points de contrôle. Les fonctions $N_{i, k+1}$ pour $i \in \llbracket -k, g \rrbracket$ sont les fonctions de la base des B-splines [3]. L'influence d'un point de contrôle est locale et nous notons $\text{infl}(P_{i,j})$ la zone d'influence du point de contrôle $P_{i,j}$:

$$\text{infl}(P_{i,j}) = [\lambda_j, \lambda_{j+k+1}] \times [\mu_i, \mu_{i+l+1}] \quad (1)$$

Un problème de minimisation. L'ajustement d'une surface peut se ramener à un problème de minimisation visant à déterminer les points de contrôle P^* tels que :

$$P^* = \arg \min_P D(P) + R_\alpha(P) \quad (2)$$

Le terme D d'attache aux données quantifie l'écart entre la surface et les données. Dans cet article, nous utilisons la moyenne des carrés des distances euclidiennes :

$$D(P) = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (s(j, i) - Z_{i,j})^2 \quad (3)$$

Le terme R_α de régularisation mesure la régularité de la surface. Il dépend d'un paramètre $\alpha \in]0, \infty[$ qui contrôle l'importance donnée à la régularisation. L'énergie de torsion³ est souvent utilisée pour ce terme. Ce choix ne permet pas d'exploiter pleinement la disposition des données sur une grille et la nature tensorielle des B-splines. Nous

2. Par abus de langage, de telles splines sont souvent appelées *produit tensoriel de B-splines* ou, plus simplement encore, *B-splines*.

3. L'énergie de torsion est définie comme l'intégrale de la somme des dérivées secondes au carré.

1. Les illustrations de cet article sont plus claires en couleur.

proposons donc un terme de régularisation paliant ces défauts. Ce terme pénalise les fortes courbures de la surface ajustée. Il est défini par :

$$R_\alpha(\mathbf{P}) = \alpha \sum_{i=1}^p \sum_{j=1}^q \left(\frac{\partial^2 s(a_i, b_j)}{\partial x^2} \right)^2 + \alpha \sum_{i=1}^p \sum_{j=1}^q \left(\frac{\partial^2 s(a_i, b_j)}{\partial y^2} \right)^2 + \alpha^2 \sum_{i=1}^p \sum_{j=1}^q \left(\frac{\partial^4 s(a_i, b_j)}{\partial x^2 \partial y^2} \right)^2 \quad (4)$$

où $a_i \in A$ et $b_j \in B$ avec $A = \{a_1, \dots, a_p\}$ et $B = \{b_1, \dots, b_q\}$. $A \times B$ définit ce que nous appelons la *grille de régularisation*.

Lorsque le paramètre de régularisation α tend vers 0, la surface ajustée tend à interpoler les données ; dans ce cas, de nombreuses oscillations erratiques ont tendance à apparaître. Au contraire, une valeur élevée pour α conduit à une surface très régulière mais qui ne reflète pas forcément très bien les données. Lorsque $\alpha \rightarrow \infty$, la surface tend vers le plan ajustant les données par moindres carrés.

Les termes D et R_α peuvent s'écrire en notation matricielle de la manière suivante :

$$D(\mathbf{P}) = \frac{1}{mn} \|\mathbf{D} \mathbf{vec}(\mathbf{P}) - \mathbf{vec}(\mathbf{Z})\|_2^2 \quad (5)$$

$$R_\alpha(\mathbf{P}) = \|\mathbf{R}_\alpha \mathbf{vec}(\mathbf{P})\|_2^2 \quad (6)$$

$$\text{où } \mathbf{D} = \mathbf{F} \otimes \mathbf{G} \quad \text{et} \quad \mathbf{R}_\alpha = \begin{bmatrix} \alpha(\mathbf{F}'' \otimes \mathbf{G}) \\ \alpha(\mathbf{F} \otimes \mathbf{G}'') \\ \alpha^2(\mathbf{F}'' \otimes \mathbf{G}'') \end{bmatrix}$$

$$F_{i,j} = N_{j-k-1, k+1}(i) \quad i \in \llbracket 1, n \rrbracket, j \in \llbracket 1, g+k+1 \rrbracket$$

$$F''_{i,j} = N''_{j-k-1, k+1}(a_i) \quad i \in \llbracket 1, p \rrbracket, j \in \llbracket 1, g+k+1 \rrbracket$$

$$G_{i,j} = N_{j-l-1, l+1}(j) \quad i \in \llbracket 1, m \rrbracket, j \in \llbracket 1, h+l+1 \rrbracket$$

$$G''_{i,j} = N''_{j-l-1, l+1}(b_i) \quad i \in \llbracket 1, q \rrbracket, j \in \llbracket 1, h+l+1 \rrbracket$$

Par conséquent, l'ajustement de surface revient à résoudre le problème de moindres carrés linéaires suivant :

$$\mathbf{P}^* = \arg \min_{\mathbf{P}} \left\| \begin{bmatrix} \mathbf{D} \\ \mathbf{R}_\alpha \end{bmatrix} \mathbf{vec}(\mathbf{P}) - \begin{bmatrix} \mathbf{vec}(\mathbf{Z}) \\ 0 \end{bmatrix} \right\|_2^2 \quad (7)$$

En considérant comme connu le paramètre de régularisation α , la solution d'un tel problème de minimisation est donnée en écrivant les équations normales [7] ; on obtient :

$$\mathbf{vec}(\mathbf{P}^*) = (\mathbf{D}^T \mathbf{D} + \mathbf{R}_\alpha^T \mathbf{R}_\alpha)^{-1} \mathbf{D}^T \mathbf{vec}(\mathbf{Z}) \quad (8)$$

En s'inspirant du travail réalisé dans [3], ce dernier calcul peut être fortement optimisé en utilisant les propriétés du produit de Kronecker :

$$\mathbf{P}^* = (\mathbf{G}^T \mathbf{G} + \alpha \mathbf{G}''^T \mathbf{G}'')^{-1} \mathbf{G}^T \mathbf{Z} \mathbf{F} (\mathbf{F}^T \mathbf{F} + \alpha \mathbf{F}''^T \mathbf{F}'')^{-1} \quad (9)$$

Le calcul de la formule (9) est plus rapide que celui de (8) car les matrices intervenant dans (9) sont plus petites que celles intervenant dans (8) : de l'ordre de g^2 (ou h^2) coefficients pour (9) contre $g^2 h^2$ coefficients pour (8). Si l'on suppose que la complexité du calcul de (8) est en $\mathcal{O}(\nu)$, alors celle de (9) est en $\mathcal{O}(\nu^{\frac{1}{2}})$.

Choix du paramètre de régularisation. Afin d'obtenir une méthode d'ajustement de surface automatique, il est indispensable de déterminer le paramètre de régularisation α . De nombreuses méthodes ont été proposées pour sélectionner ce paramètre comme, par exemple, la validation croisée [4] ou le critère de la L-tangent norm [5]. Dans cet article, nous utilisons une adaptation du *principe de divergence* (« discrepancy principle ») [8].

Considérons que l'ensemble des points de données est un échantillon bruité d'une certaine fonction théorique \underline{f} c'est-à-dire que $Z_{i,j} = \underline{f}(j, i) + E_{i,j}$ où E représente le bruit. Considérons, de plus, que la distribution du bruit est unimodale et que son écart type est connu ($\sigma \stackrel{\text{def}}{=} \text{sd}(E)$). Soit \mathcal{S} l'ensemble des B-splines bidimensionnelles solutions de l'équation (7) (pour \mathbf{Z} fixé). L'ajustement de surface consiste à trouver une fonction $f \in \mathcal{S}$ telle que :

$$f = \arg \min_{s \in \mathcal{S}} \|s - \underline{f}\| \quad (10)$$

Le principe de divergence consiste à choisir la fonction $f \in \mathcal{S}$ telle que $\text{sd}(\mathbf{S} - \mathbf{Z})$ soit le plus proche possible de σ avec $S_{i,j} = f(j, i)$. Comme généralement $\underline{f} \notin \mathcal{S}$, l'égalité entre $\text{sd}(\mathbf{S} - \mathbf{Z})$ et σ n'est pas toujours possible⁴. Le paramètre de régularisation est donc choisi comme solution du problème de minimisation suivant :

$$\alpha = \arg \min_{\alpha \in \mathbb{R}_+} (\text{sd}(\mathbf{S}^*(\alpha) - \mathbf{Z}) - \sigma)^2 \quad (11)$$

où $S_{i,j}^*(\alpha) = s_\alpha(j, i)$ et s_α est la B-spline obtenue en résolvant l'équation (7) pour une valeur de α fixée. Comme la quantité $\text{sd}(\mathbf{S}^* - \mathbf{Z})$ croît avec α , le critère optimisé dans (11) ne possède qu'un seul minimum. Le problème (11) est optimisé en utilisant la méthode « golden section search ».

En pratique, l'écart type des erreurs sur les données est rarement connu. Nous proposons une heuristique permettant d'estimer l'écart type à partir de données bruitées. Soit $\mathcal{P}_{i,j}$ la fonction représentant le plan obtenu par régression linéaire multiple à partir des données $\{(v, u) \leftrightarrow Z_{u,v} | u \in \llbracket i-1, i+1 \rrbracket \text{ et } v \in \llbracket j-1, j+1 \rrbracket\}$. L'estimation σ_e de l'écart type de l'ensemble des points est donnée par :

$$\sigma_e^2 = \frac{1}{(m-2)(n-2)} \sum_{i=2}^{m-1} \sum_{j=2}^{n-1} (\mathcal{P}_{i,j}(j, i) - Z_{i,j})^2 \quad (12)$$

Choix des autres paramètres. Dans cet article nous considérons des B-splines bicubiques, c'est-à-dire que $k = 3$ et $l = 3$. De plus, nous utilisons des vecteurs nodaux coïncidents L et M tels que :

$$\begin{aligned} \lambda_{-k} = \dots = \lambda_0 = \frac{1}{2}, \quad \lambda_{g+1} = \dots = \lambda_{g+k+1} = n + \frac{1}{2} \\ \lambda_i = i + \frac{1}{2} \quad \text{pour } i \in \llbracket 1, g \rrbracket \\ \mu_{-l} = \dots = \mu_0 = \frac{1}{2}, \quad \mu_{h+1} = \dots = \lambda_{h+l+1} = m + \frac{1}{2} \\ \mu_j = j + \frac{1}{2} \quad \text{pour } j \in \llbracket 1, h \rrbracket \end{aligned}$$

4. Par exemple, si le degré de liberté du modèle de fonction choisi est trop faible, le plus petit écart type sur les erreurs de reconstruction (obtenu pour $\alpha = 0$) sera supérieur à l'écart type attendu.

Enfin, la grille de régularisation $A \times B$ est définie par :

$$a_i = \frac{i}{2} \quad \text{et} \quad b_j = \frac{j}{2} \quad (13)$$

pour $i \in \llbracket 1, 2n + 1 \rrbracket$ et $j \in \llbracket 1, 2m + 1 \rrbracket$.

3 Ajustements locaux

Afin de prendre en compte les spécificités des données, nous proposons une nouvelle méthode d'ajustement de surface dont les étapes sont les suivantes :

Segmentation. Les données sont dans un premier temps segmentées afin de prendre en compte la variabilité du bruit en fonction de la profondeur.

Boîtes englobantes. Chaque classe de la segmentation est ensuite plongée dans un domaine rectangulaire afin de pouvoir exploiter les techniques rapides d'ajustement à partir de données disposées sur une grille.

Cartes locales de profondeur. Toujours dans l'objectif d'utiliser les techniques rapides d'ajustement, les données initiales contenues dans chaque boîte englobante sont complétées afin de « remplir » le domaine.

Ajustements locaux. Une surface est ajustée sur chaque carte locale de profondeur en utilisant la méthode décrite à la section précédente.

Regroupement. En utilisant les propriétés d'influence locale des points de contrôle des B-splines, les ajustements locaux sont regroupés pour former une seule et unique surface.

La figure 2 illustre les étapes de notre approche à partir des données de la figure 1.

3.1 Segmentation

La première étape de notre méthode consiste à segmenter les données en fonction des profondeurs. Pour cela, nous utilisons une adaptation de la méthode de segmentation basée sur l'algorithme α -expansion [9, 10]. Les détails de la mise en place de cette méthode sont décrits en annexe A.1. Le résultat de la segmentation est composé d'un ensemble de r étiquettes $\mathcal{E} = \llbracket 1, r \rrbracket$ et d'un ensemble de r classes $\mathcal{C}^{(t)}$ pour $t \in \llbracket 1, r \rrbracket$. La classe $\mathcal{C}^{(t)}$ contient l'ensemble des pixels $(i, j) \in \llbracket 1, m \rrbracket \times \llbracket 1, n \rrbracket$ étiquetés par la valeur t . De plus, la classe $\mathcal{C}^{(t)}$ est 8-connexe. On note $C^{(t)}$ la matrice telle que $C_{i,j}^{(t)} = 1$ si $(i, j) \in \mathcal{C}^{(t)}$ et 0 sinon.

3.2 Boîtes englobantes

De manière générale, les classes $\{\mathcal{C}^{(t)}\}_{t \in \llbracket 1, r \rrbracket}$ issues de la segmentation ne sont pas rectangulaires. Or, les données à ajuster doivent être disposées sur une grille pour pouvoir exploiter les techniques rapides d'ajustement de surfaces. Pour cela, nous définissons pour chaque classe $\mathcal{C}^{(t)}$ une *boîte englobante* $\mathcal{B}^{(t)}$ qui est, intuitivement, un sous-domaine rectangulaire de $[\lambda_{-k}, \lambda_{g+k+1}] \times [\mu_{-l}, \mu_{h+l+1}]$ encadrant les données de la classe $\mathcal{C}^{(t)}$ et de taille minimale. Plus formellement, $\mathcal{B}^{(t)}$ est le plus petit domaine rectangulaire $\llbracket \beta^{(t)}, \delta^{(t)} \rrbracket \times \llbracket \gamma^{(t)}, \eta^{(t)} \rrbracket$ (avec $\beta^{(t)}, \delta^{(t)} \in L$ et $\gamma^{(t)}, \eta^{(t)} \in M$) qui contient le domaine (non rectangulaire)

$\mathcal{I}^{(t)}$ défini comme suit :

$$\mathcal{I}^{(t)} = \bigcup_{\substack{(i,j) \text{ t.q.} \\ (u,v) \in \mathbf{infl}(P_{i,j}) \\ \text{pour } (u,v) \in \mathcal{C}^{(t)}}} \mathbf{infl}(P_{i,j}) \quad \text{pour } t \in \llbracket 1, r \rrbracket \quad (14)$$

3.3 Cartes locales de profondeur

Toujours dans l'objectif de pouvoir exploiter les techniques rapides d'ajustement de surfaces sur grille, nous avons besoin de données sur *toute* la grille. Nous définissons donc pour chaque classe $\{\mathcal{C}^{(t)}\}_{t \in \llbracket 1, r \rrbracket}$ des *cartes locales de profondeur*. Pour $t \in \llbracket 1, r \rrbracket$, la carte locale de profondeur $Z^{(t)} \in \mathbb{R}^{m \times n}$ est définie par :

$$Z_{i,j}^{(t)} = \begin{cases} Z_{i,j} & \text{si } (i, j) \in \mathcal{C}^{(t)} \\ 0 & \text{si } (i, j) \notin \mathcal{B}^{(t)} \\ \hat{Z}_{i,j}^{(t)} & \text{sinon} \end{cases} \quad (15)$$

Les valeurs $\hat{Z}_{i,j}^{(t)}$ sont calculées par une méthode de propagation des profondeurs situées sur les bordures de la classe $\mathcal{C}^{(t)}$. Le détail de cette opération est donné par l'algorithme 1 (annexe A.2).

3.4 Ajustements locaux

L'étape suivante de notre méthode consiste à ajuster une B-spline bidimensionnelle sur chacune des r cartes locales de profondeur $Z^{(t)}$. Cet ajustement est effectué avec la méthode décrite dans la section 2 et en utilisant les vecteurs nodaux et la grille de régularisation suivants :

$$\begin{aligned} \bar{M}^{(t)} &= M \cap [\beta^{(t)}, \delta^{(t)}] & \bar{A}^{(t)} &= A \cap [\beta^{(t)}, \delta^{(t)}] \\ \bar{L}^{(t)} &= L \cap [\gamma^{(t)}, \eta^{(t)}] & \bar{B}^{(t)} &= B \cap [\gamma^{(t)}, \eta^{(t)}] \end{aligned}$$

Le résultat de cette étape est constitué d'un ensemble de r matrices de points de contrôle $P^{(t)} \in \mathbb{R}^{(g+k+1) \times (h+l+1)}$. Notons que $P_{i,j}^{(t)} = 0$ s'il n'existe pas $(u, v) \in \mathcal{B}^{(t)}$ tel que $(u, v) \in \mathbf{infl}(P_{i,j})$.

Comme le bruit est homogène sur chacune des classes, nous utilisons le principe de divergence décrit à la section 2 pour déterminer les paramètres de régularisation $\alpha^{(t)}$. De plus, cette approche permet de ne prendre en compte qu'une partie des données d'une carte locale de profondeur $Z^{(t)}$ (en l'occurrence, les données de la classe $\mathcal{C}^{(t)}$). Cela ne serait pas possible en utilisant une méthode comme la validation croisée.

3.5 Regroupement

La dernière étape de notre méthode consiste à regrouper les ajustements locaux afin d'obtenir une seule et unique surface ajustant la carte de profondeur initiale Z . Pour cela, nous construisons une B-spline en prenant les points de contrôle calculés lors des ajustements locaux. Pour $t \in \llbracket 1, r \rrbracket$, soit $Q^{(t)} \in \mathbb{R}^{(g+k+1) \times (h+l+1)}$ la matrice indiquant les points de contrôle « intéressants » par rapport à la classe $\mathcal{C}^{(t)}$:

$$Q_{i,j}^{(t)} = \begin{cases} 1 & \text{si } \exists (u, v) \in \mathcal{C}^{(t)} \text{ t.q. } (u, v) \in \mathbf{infl}(P_{i,j}) \\ 0 & \text{sinon} \end{cases}$$

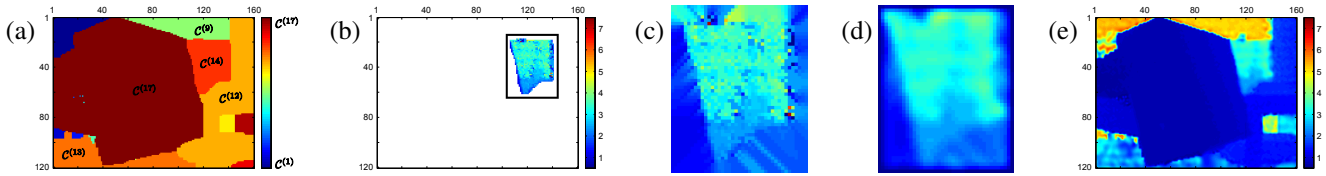


Figure 2 – (a) Segmentation. (b) Boîte englobante pour la classe $C^{(14)}$. (c) Carte locale de profondeur extrapolée à partir des données de la classe $C^{(14)}$. (d) Ajustement local. (e) Regroupement des ajustements locaux.

Afin qu’il n’y ait pas de conflits entre les points de contrôle provenant des différents ajustements locaux, nous réduisons l’ensemble des points de contrôle intéressants en érodant la matrice binaire $Q^{(t)}$:

$$\hat{Q}^{(t)} = Q^{(t)} \ominus S_2 \quad (16)$$

où S_2 est l’élément structurant suivant (pour le cas de B-splines bicubiques) :

$$S_2 = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (17)$$

Finalement, nous obtenons une B-spline globale ajustant la carte de profondeur Z dont la matrice P^* des points de contrôle est donnée par :

$$P^* = \sum_{t=1}^r \bar{P}^{(t)} \quad (18)$$

$$\text{avec } \bar{P}_{i,j}^{(t)} = \begin{cases} P_{i,j}^{(t)} & \text{si } \hat{Q}_{i,j}^{(t)} = 1 \\ 0 & \text{sinon} \end{cases} \quad (19)$$

4 Résultats expérimentaux

Estimateur d’écart type. Pour cette expérience, nous utilisons des surfaces dont les formules analytiques sont générées aléatoirement comme combinaisons linéaires et compositions de fonctions élémentaires (sinus, exponentielles, polynômes). Ces surfaces sont ensuite échantillonnées sur une grille de 120×160 points en ajoutant un bruit gaussien centré d’écart type σ . L’écart type estimé σ_e est comparé à la valeur attendue en calculant l’erreur relative. Nous avons essayé notre estimateur pour différents niveaux de bruits (σ variant de 0,01% à 0,21% de l’amplitude des données). En répétant l’estimation sur 100 surfaces différentes pour chaque niveau de bruit, nous constatons que l’erreur relative est inférieure à 6%.

Méthode globale et méthode locale. Nous comparons ici la surface obtenue par notre méthode locale avec le résultat d’une approche globale. Pour cela nous reprenons l’exemple représentatif utilisé jusque là dans cet article. Dans la mesure où le principe de divergence n’est applicable que si le bruit est de même nature pour l’ensemble des données, nous utilisons la validation croisée généralisée [4] afin de déterminer l’importance de la régularisation dans le cas de l’ajustement global. Grâce à la régularisation adaptative de notre méthode (voir table 1), nous pouvons constater sur les vignettes 3a et 3c que les zones fortement bruitées sont plus régularisées que les zones moins

bruitées. De plus, malgré les régularisations localement importantes, les détails du premier plan sont conservés (vignette 3c). Les vignettes 3b et 3d montrent que l’utilisation d’une méthode globale entraîne une régularisation trop importante sur les détails du premier plan et sur les discontinuités de la scène.

$C^{(t)}$	$\alpha^{(t)}$
$C^{(9)}$	0,2394
$C^{(12)}$	0,0325
$C^{(13)}$	1,8936
$C^{(17)}$	0,0138

Tableau 1 – Quelques exemples de paramètres de régularisation obtenus par le principe de divergence pour les données de la figure 2.

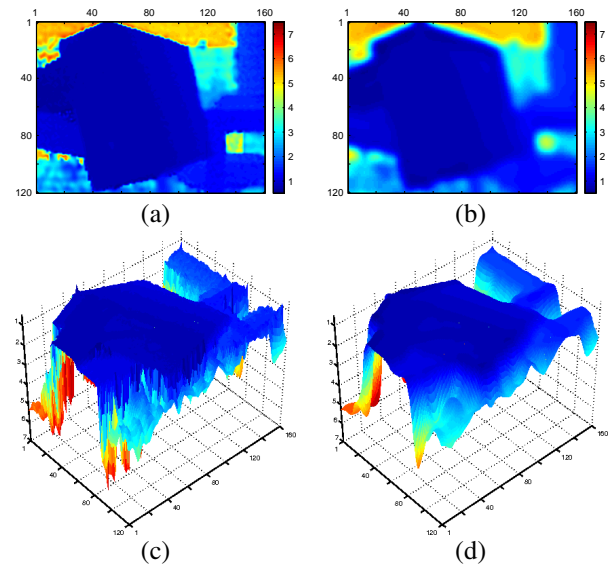


Figure 3 – Comparaison entre (a) notre méthode et (b) la méthode globale. (c) et (d) Visualisation 3D des cartes (a) et (b).

Un autre exemple de comparaison entre notre méthode et une approche globale est donnée en figure 4.

5 Conclusion

Nous avons présenté une nouvelle approche permettant d’ajuster une surface globale sur des données de profondeur hétérogènes. Même si notre approche fonctionne, certaines améliorations pourraient être apportées. Par exemple, au lieu d’effectuer la segmentation sur les profondeurs, nous pourrions segmenter la scène en fonction des différents niveaux de bruit. Il pourrait aussi être judicieux d’utiliser des informations supplémentaires (comme les contours d’objets) pour guider le processus d’ajustement. En effet, outre les données de profondeur, les camé-

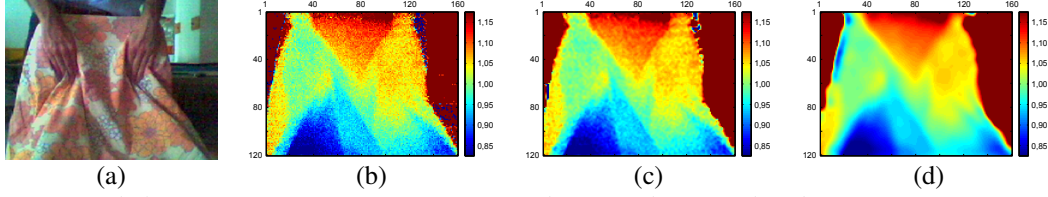


Figure 4 – (a) Aperçu de la scène pris par un capteur CMOS. (b) Carte de profondeur brute. (c) Ajustement par la méthode proposée. (d) Ajustement global. L'échelle des couleurs est choisie pour mettre en évidence le premier plan de la scène.

ras TdV peuvent fournir une image en niveau de gris. Il serait aussi intéressant de construire une méthode permettant de déterminer automatiquement le nombre de classes à utiliser pour effectuer la segmentation de la scène.

Annexe

A.1 Segmentation Afin d'appliquer notre méthode d'ajustement, la carte de profondeur Z doit être segmentée en r classes $\{\mathcal{C}^{(t)}\}_{t \in \llbracket 1, r \rrbracket}$. Soit $\mathcal{J} = \llbracket 1, m \rrbracket \times \llbracket 1, n \rrbracket$ l'ensemble des pixels de la carte de profondeur. La segmentation consiste à classer chaque pixel de \mathcal{J} dans une et une seule des classes $\mathcal{C}^{(t)}$ pour $t \in \llbracket 1, r \rrbracket$. De plus, nous désirons que (1) la segmentation soit spatialement cohérente et que (2) les classes soient 8-connexes. Pour cela, nous utilisons l'algorithme α -expansion proposé dans [9, 10]. Avec cet algorithme, la segmentation est obtenue en tant que solution d'un problème de minimisation de la forme :

$$\min_{\{\mathcal{C}^{(1)}, \dots, \mathcal{C}^{(s)}\}} E_{data} + E_{reg} \quad (20)$$

où s est un entier tel que $s \leq r$. E_{data} est un terme d'attache aux données c'est-à-dire un terme mesurant la qualité d'une affectation de tous les pixels à certaines classes. E_{reg} est un terme favorisant la cohérence spatiale.

Nous avons choisi comme terme d'attache aux données la vraisemblance qu'une profondeur $Z_{i,j}$ appartienne à la classe $\mathcal{C}^{(t)}$. En considérant que la distribution des données sur chacune des s classes suit une loi normale de paramètre $\theta^{(t)} = (\mu^{(t)}, \sigma^{(t)})$, E_{data} est donné par :

$$E_{data} = \sum_{t \in \llbracket 1, s \rrbracket} \sum_{(i,j) \in \mathcal{J}} D^{(t)}(i,j) \quad (21)$$

$$\text{où } D^{(t)}(i,j) = -\log(L((i,j) \in \mathcal{C}^{(t)} | \theta^{(t)})) \quad (22)$$

$$\text{et } L((i,j) \in \mathcal{C}^{(t)} | \theta^{(t)}) \propto \exp\left(-\frac{(Z_{i,j} - \mu^{(t)})^2}{2(\sigma^{(t)})^2}\right) \quad (23)$$

Les paramètres $\{\theta^{(t)}\}_{t \in \llbracket 1, s \rrbracket}$ de chacune des classes sont calculés en utilisant l'algorithme k -means sur la carte de profondeur Z (le nombre s étant déterminé manuellement). Pour le terme de régularisation E_{reg} , nous utilisons le modèle de Potts ($\mathcal{N}(\mathbf{p})$ désignant l'ensemble des pixels appartenant au 8-voisinage du pixel \mathbf{p}) :

$$E_{reg} = \sum_{\mathbf{p} \in \mathcal{J}} \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{p})} V(\mathbf{p}, \mathbf{q}) \quad (24)$$

$$\text{où } V(\mathbf{p}, \mathbf{q}) = \begin{cases} 0 & \text{si } \mathbf{p} \in \mathcal{C}^{(t)} \text{ et } \mathbf{q} \in \mathcal{C}^{(t)} \\ c & \text{si } \mathbf{p} \in \mathcal{C}^{(t)} \text{ et } \mathbf{q} \in \mathcal{C}^{(t')}, t \neq t' \end{cases} \quad (25)$$

La valeur $c \in \mathbb{R}_+$ détermine l'importance accordée à la régularisation ; nous prenons $c = 25$. Enfin, la dernière étape de la segmentation consiste à prendre les r composantes connexes des s classes $\{\mathcal{C}^{(t)}\}_{t \in \llbracket 1, s \rrbracket}$.

A.2 Calcul des cartes locales de profondeur

Algorithme 1 : Calcul des cartes $\tilde{Z}^{(t)}$

Entrées : $\mathcal{B}^{(t)}, Z, \mathcal{C}^{(t)}$ Sortie : $\tilde{Z}^{(t)}$

Données : $S = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

$A \leftarrow \mathcal{C}^{(t)}$

$T \leftarrow (A \oplus S) \setminus A$

$\mathcal{T} \leftarrow \{(i,j) \in \mathcal{B}^{(t)} \mid \mathbb{T}_{i,j} = 1\}$

tant que $T \neq \emptyset$ **faire**

pour $(i,j) \in \mathcal{T}$ **faire**

$\mathcal{V} \leftarrow \{(u,v) \mid |u-i| \leq 1 \text{ et } |v-j| \leq 1\}$

$z \leftarrow 0; n \leftarrow 0$

pour $(u,v) \in \mathcal{V} \cap \mathcal{C}^{(t)}$ **faire**

$z \leftarrow z + Z_{u,v}; n \leftarrow n + 1$

$\tilde{Z}_{i,j}^{(t)} \leftarrow z/n$

$A \leftarrow A + T$

$T \leftarrow (A \oplus S) \setminus A$

$\mathcal{T} \leftarrow \{(i,j) \in \mathcal{B}^{(t)} \mid \mathbb{T}_{i,j} = 1\}$

L'algorithme ci-contre complète les cartes locales de profondeur (section 3.3) en propageant de manière gloutonne les valeurs situées en bordure des classes issues de la segmentation.

Références

- [1] Z. Xu, R. Schwarte, H. Heinol, B. Buxbaum, et T. Ringbeck. Smart pixel - Photonic Mixer Device (PMD). Rapport technique, PMDTec, 2005.
- [2] A. Swadzba, B. Liu, J. Penne, O. Jesorsky, et R. Kompe. A comprehensive system for 3D modeling from range images acquired from a 3D ToF sensor. In *International Conference on Computer Vision Systems*, 2007.
- [3] P. Dierckx. *Curve and surface fitting with splines*. Oxford University Press, Inc., 1993.
- [4] G. Wahba. *Spline models for observational data*. Society for Industrial and Applied Mathematics, 1990.
- [5] F. Brunet, A. Bartoli, R. Malgouyres, et N. Navab. L-Tangent Norm : A low computational cost criterion for choosing regularization weights and its use for range surface reconstruction. In *Fourth International Symposium on 3D Data Processing, Visualization and Transmission*, 2008.
- [6] D. Falie et V. Buzuloiu. Noise characteristics of 3D Time-of-Flight cameras. In *International Symposium on Signals, Circuits & Systems*, 2007.
- [7] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, 1996.
- [8] E. Schock. Parameter choice by discrepancy principles for the approximate solution of ill-posed problems. *Integral Equations and Operator Theory*, 7 :895–898, 1984.
- [9] Y. Boykov, O. Veksler, et R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence*, 23 :1222–1239, 2001.
- [10] R. Zabih et V. Kolmogorov. Spatially coherent clustering using graph cuts. In *Computer Vision and Pattern Recognition*, 2004.