



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Recalage géométrique avec plusieurs prototypes

Jean-Philippe Tarel

N° 2988

Septembre 1996

_____ THÈME 3 _____

A large, light gray, stylized letter 'R' that serves as a background for the text 'Rapport de recherche'. The 'R' is positioned to the left of the text, with its vertical stroke extending downwards and its curved stroke extending to the right.

*Rapport
de recherche*



Recalage géométrique avec plusieurs prototypes

Jean-Philippe Tarel *

Thème 3 — Interaction homme-machine,
images, données, connaissances
Projet Syntim

Rapport de recherche n° 2988 — Septembre 1996 — 62 pages

Résumé : Nous décrivons un cadre général pour l'interprétation précise et robuste d'un ensemble de points par plusieurs prototypes déformables. Cette méthode est basée sur l'unification de deux algorithmes : une généralisation de l'algorithme "Iterative Closest Point" (GICP) à différents types de transformations pour des tâches de recalage, et un algorithme de classification floue (FCM) pour traiter plusieurs prototypes.

Notre algorithme converge de façon monotone vers le plus proche minimum local d'une fonction de coût au moindre carré, et les expériences montrent que la convergence est rapide dans les premières étapes. En conséquence, nous avons proposé un schéma pour choisir la position initiale des prototypes pour qu'ils convergent vers une solution "intéressante". La méthode présentée est très générique et peut être appliquée :

- à des prototypes dans un espace de dimension p quelconque,
- à différentes formes de prototypes comme les polyèdres, les quadriques, les fonctions polynômiales, les snakes,
- à de nombreux types de déformations comme les déplacements rigides, les similitudes, les affinités et les homographies.

* E-mail : Jean-Philippe.Tarel@inria.fr

Ainsi, notre méthode a un grand nombre d'applications en recalage avec un modèle idéal connu a priori, c'est-à-dire pour interpréter des données 2D et 3D obtenues par des capteurs calibrés ou non. Des résultats expérimentaux illustrent quelque'un des possibilités de notre approche.

Mots-clé : géométrie 2D et 3D, reconnaissance par morceaux, interprétation avec modèle, recalage géométrique, classification floue, ICP.

(Abstract: pto)

INRIA

Geometric registration with several prototypes

Abstract: We describe a general-purpose method for the accurate and robust interpretation of a data set of p -dimensional points by several deformable prototypes. This method is based on the fusion of two algorithms: a Generalization of the Iterative Closest Point (GICP) to different types of deformations for registration purposes, and a fuzzy clustering algorithm (FCM).

Our method always converges monotonically to the nearest local minimum of a mean-square distance metric, and experiments show that the convergence is fast during the first few iterations. Therefore, we propose a scheme for choosing the initial solution to converge to an "interesting" local minimum. The method presented is very generic and can be applied:

- to shapes or objects in a p -dimensional space,
- to many shape patterns such as polyhedra, quadrics, polynomial functions, snakes,
- to many possible shape deformations such as rigid displacements, similitudes, affine and homographic transforms.

Consequently, our method has important applications in registration with an ideal model prior to shape inspection, i.e. to interpret 2D or 3D sensed data obtained from calibrated or uncalibrated sensors. Experimental results illustrate some capabilities of our method.

Key-words: 2D and 3D geometry, recognition by parts, model-based interpretation, geometric registration, fuzzy clustering, ICP.

1 Introduction

Accumuler, par des reconstructions suivant différents points de vue, des informations géométriques 3D sur un même objet n'est pas suffisant pour obtenir un modèle de celui-ci. Les différentes reconstructions recalées entre elles contiennent en effet souvent des parties redondantes qu'il faut fusionner et des parties manquantes qu'il faut interpoler. Modéliser consiste donc à construire une représentation plus abstraite de ces données 3D ou 2D.

Une approche pour traiter ce problème qui combine la mise en correspondance des données et le recalage est de chercher à recouvrir toutes les données par un ensemble d'instance d'un même prototype paramétré. Comme notre algorithme de reconstruction produit des facettes [46], il est pertinent d'obtenir une modélisation sous la forme d'un ensemble de prototypes qui sont des parallélépipèdes.

Dans le cas 3D, on peut distinguer deux classes d'algorithmes pour résoudre le problème du recalage avec appariement partiel : ceux qui remettent itérativement en cause les appariements et ceux qui procèdent par accumulation. Mais, même si la description des objets peut être géométriquement complexe, un seul modèle ou prototype peut être utilisé à la fois, par ces différents types de méthodes. En particulier, c'est le cas de la méthode que nous avons décrite [47, 48].

Contrairement au cas 3D, en 2D, une approche populaire d'interprétation des données est fondée sur l'utilisation des méthodes de classification. Ainsi, les prototypes utilisés sont très simples : ce sont des lignes ou des ellipses. Mais en revanche, plusieurs instances du prototype sont utilisées en même temps. En particulier, les méthodes de classification floue ont montré leurs capacités à réaliser des classifications géométriques [34, 15, 35].

Ces deux démarches, n'ont pas de raison d'être spécifiques au 2D ou au 3D et sont de plus fortement complémentaires. Mais comment partitionner au mieux les données pour que sur chaque partie, le prototype déformable se superpose bien aux données? Des travaux dans ce sens utilisent comme prototypes les super-quadriques [27, 21, 41], des surfaces algébriques [49] ou plus simplement les quadriques [24].

S'il est impossible de séparer partitionnement et estimation dans ce type de problème [2], alors, la classification floue fournit un cadre intéressant pour le poser. De fait, nous avons formalisé en dimension quelconque ce problème, dans une description unificatrice qui intègre non seulement les avantages des méthodes de recalage

développées dans le contexte de la vision par ordinateur 3D mais aussi ceux complémentaires des algorithmes de reconnaissance des formes 2D.

L'unification [45] des algorithmes FCM et ICP est très puissante et permet de produire une classe d'algorithmes très générique. En effet, cette approche, que nous présentons dans la section 2, peut être utilisée pour générer de nombreux algorithmes propres à un type de modèle dans l'espace de dimension p de son choix. Nous décrivons dans les sections suivantes trois applications de cette approche. Dans la section 3, la généralisation de l'algorithme ICP (GICP) à des transformations autres que le déplacement rigide comme les similitudes, les homographies, nous permet de concevoir un algorithme de recalage à plusieurs prototypes. Dans la section suivante 4, nous avons appliqué notre approche à la segmentation d'images de niveaux de gris ou en couleurs, comme aux cartes de profondeur. Enfin, dans la dernière section nous présentons, toujours dans le même esprit, un algorithme pour recaler finement et de façon coordonnée plusieurs contours actifs.

2 L'approche Modélisation Multi-Prototypes (MMP)

Pour décrire plus en détails notre approche, nous avons choisi de décrire, dans la prochaine section, un algorithme de recouvrement par des formes géométriques simples. C'est ce problème initial qui a motivé cette étude. Mais, nous voulons insister sur le fait que cet algorithme n'est que l'application d'un principe d'algorithme plus général. Ce principe consiste à formuler l'appariement, problème de nature combinatoire a priori, comme un problème d'estimation par minimisation sur un support continu, grâce à l'utilisation des degrés d'appartenance introduits dans le cadre des ensembles flous.

On peut résumer le principe de cette méthode par :

- Une étape **d'initialisation** qui doit être astucieuse pour faciliter au plus le travail des deux étapes suivantes. Cette étape est spécifique de l'objectif de chaque problème et du type de données.
- Une étape **d'appariement** où on recherche, pour chaque donnée X_i ($i = 1, \dots, n$), le plus proche voisin X'_{ik} dans le prototype k (avec éventuellement un prototype qui définit une classe de rejet). Ceci permet d'associer à chaque donnée un correspondant potentiel fonction de l'état courant du prototype.

Ensuite, la formule suivante permet de déterminer le degré d'appartenance de chaque donnée à chaque prototype en se fondant sur leur proximité :

$$u_{ik} = \frac{1}{\sum_{j=1}^m \left(\frac{d(X_i, X'_{ik})}{d(X_i, X'_{ij})} \right)^{\frac{2}{m-1}}}$$

- Une étape de **recalage** de chaque prototype k sur les données. Chaque donnée i est pondérée par son degré d'appartenance u_{ik} associée à la classe. Ce prototype peut être une forme géométrique simple (section 3.4), une fonction sur \mathbb{R}^2 (section 4), ou une structure géométrique déformable plus complexe (section 5).
- Une étape **terminale**, qui par un test sur la stabilité des caractéristiques des prototypes obtenus, permet de savoir s'il faut itérer ou non les deux étapes précédentes.

3 Application au recouvrement par des formes géométriques simples

Dans cette partie, nous présentons une version de l'algorithme MMP, quand on choisit pour prototype une forme géométrique de référence simple qui peut se déformer suivant une transformation géométrique usuelle. La méthode explique un ensemble donné de points comme le recouvrement de c primitives géométriques qui se déforment dans un ensemble de transformations spécifiées. Mais, commençons par résoudre ce problème dans le cas d'une unique primitive.

3.1 Généralisation de la méthode ICP (GICP)

Une généralisation naturelle de l'algorithme ICP [5] est de remplacer le déplacement rigide par un autre type de transformation F , notamment une similitude, une affinité ou une homographie.

Étant donné un ensemble de points, nous supposons que nous connaissons le prototype de cet ensemble et le type de transformation qui peut exister entre la

forme modèle et les données. Par exemple, si l'ensemble échantillonne un parallélogramme, on peut choisir comme forme modèle, le carré unité dans son repère propre, et comme ensemble de déformations, les transformations affines, pour modéliser cet ensemble de points. Le but du GICP est d'obtenir la meilleure transformation au sens des moindres carrés qui superpose la forme de référence avec le jeu de données.


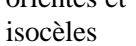





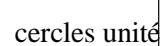
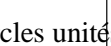




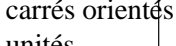


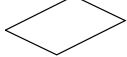

	Translation	Déplacement	Similitude	Affinité	Homographie
triangle unité centré 	triangles unités orientés et isocèles 	triangles unités isocèles 	triangles isocèles 	triangles 	triangles 
cercle unité centré 	cercles unités 	cercles unités 	cercles 	ellipses 	coniques 
carré unité centré 	carrés orientés unités 	carrés unités 	carrés 	parallélogrammes 	quadrilatères 

FIG. 1 – Différentes formes engendrées par 5 types de transformations géométriques appliquées à 3 formes de référence particulières : le cercle, le carré et le triangle.

Pour les primitives particulières suivantes, un carré unité, un cercle unité et un triangle isocèle unité dans leur système de coordonnées propres, il est intéressant de garder à l'esprit la classe des formes engendrées par l'application des transformations citées plus haut, pour entrevoir l'utilisation possible de l'algorithme GICP. Par exemple, avec le carré unité, l'ensemble des carrés, des rectangles, des parallélogrammes, et des quadrilatères peut être engendré en choisissant le type de transformation adéquat. Avec le cercle unité, c'est l'ensemble des cercles, des ellipses, et des coniques qui sont engendrés. Le tableau 1 montre en détails toutes les possibilités.

Plus généralement, comme pour l'ICP, le GICP peut travailler avec des modèles définis par des ensembles de points, de lignes, avec des courbes implicites ou paramétriques, et avec des surfaces maillées, implicites ou paramétriques.

3.1.1 L'algorithme GICP

L'algorithme GICP, où l'ensemble des n points $\{X_i\}_{1,\dots,n}$, est connu, est le suivant :

étape 0 : Fixer la forme du modèle de référence S_{ref} . F est initialisée.

étape g : Calculer le nouveau point X'_i appartenant au modèle S_{ref} qui est le plus proche du point $F(X_i)$, pour chaque i .

étape g' : Calculer la nouvelle transformation F qui minimise l'erreur aux moindres carrés entre les points X'_i et $F(X_i)$.

étape g'' : Si l'erreur aux moindres carrés est réduite d'une valeur inférieure à un seuil donné a priori qui spécifie la précision, alors il faut s'arrêter ou sinon retourner à l'**étape g**.

L'erreur aux moindres carrés minimisée par l'algorithme GICP est :

$$e_{GICP}(F, X'_i) = \sum_{i=1}^n d^2(F(X_i), X'_i) \quad \text{avec} \quad X'_i \in S_{ref} \quad (1)$$

où $d(X, X')$ est la distance euclidienne entre deux points X et X' .

Pour pouvoir généraliser l'algorithme ICP, il est donc nécessaire d'être capable d'exprimer analytiquement quelle est la transformation optimale qui minimise l'erreur aux moindres carrés entre deux ensembles de n points appariés (**étape g'**). Lorsque $\{X'_i\}_{1,\dots,n}$ est connu, l'expression de la transformation optimale qui minimise $e_{GICP}(F)$ dans l'équation (1) pour différents types de transformations est donnée dans la section suivante.

3.1.2 Estimation avec appariement

Dans cette section, nous disposons de deux ensembles de n points $\{X_i\}_{1,\dots,n}$ et $\{X'_i\}_{1,\dots,n}$, où $i = 1, \dots, n$ qui sont appariés pour un même indice i . L'ensemble

$\{X'_i\}_{1,\dots,n}$ est le transformé par F de l'ensemble $\{X_i\}_{1,\dots,n}$. Ceci se traduit par la relation suivante :

$$X'_i = F(X_i) + B_i$$

où B_i est un bruit supposé gaussien centré. Dans le cadre du recalage, les transformations F les plus utiles sont :

- une translation $F(X) = X + T$,
- un déplacement rigide, soit une rotation R combinée avec une translation $F(X) = RX + T$,
- une similitude $F(X) = sRX + T$ où s est un scalaire,
- un étirement, c'est-à-dire la combinaison d'un changement d'échelle sur chaque axe avec un déplacement rigide, soit $F(X) = RDX + T$ où D est une matrice diagonale,
- une affinité $F(X) = LX + T$, qui est une combinaison d'une transformation linéaire L et d'une translation.
- une homographie qui est représentée par \mathcal{H} , une matrice $(p+1) \times (p+1)$. L'image X' d'un point X est $\mathcal{X}' \equiv \mathcal{H}\mathcal{X}$, en coordonnées homogènes ($\mathcal{X} = \begin{pmatrix} X \\ 1 \end{pmatrix}$) avec \equiv l'égalité à un facteur près.

On remarque que ces transformations définissent des transformations de plus en plus générales. Elles vont de l'espace \mathbb{R}^p vers \mathbb{R}^p .

Étant donné deux ensembles de points, estimer au mieux la transformation entre les deux ensembles bruités par un bruit gaussien revient à minimiser la fonctionnelle suivante :

$$e(F) = \sum_{i=1}^n d^2(F(X_i), X'_i) \quad (2)$$

où $d(X, X')$ est la distance euclidienne entre les points X et X' .

Nous allons maintenant récapituler, pour les différents types de transformations, la méthode la plus générale pour estimer la meilleure transformation minimisant cette fonctionnelle (2). Nous nous sommes focalisés sur les méthodes non itératives

qui présentent l'avantage évident de ne pas nécessiter de point initial proche de la solution exacte.

Translation Nous désirons trouver la translation T qui minimise :

$$e(T) = \sum_{i=1}^n \|X_i + T - X'_i\|^2$$

Une condition nécessaire pour que cette fonction e soit minimum est que $\nabla e = 0$. Généralement, cette condition n'est pas suffisante, mais dans le cas présent, elle est nécessaire et suffisante. De plus, il y a toujours unicité de la solution. Par une simple dérivation, on obtient la meilleur translation, comme la différence des centres de gravité des deux ensembles :

$$T = \frac{1}{n} \sum_{i=1}^n X'_i - \frac{1}{n} \sum_{i=1}^n X_i = \overline{X}'_i - \overline{X}_i$$

Nous avons rappelé ce cas très simple car l'estimation de la translation peut souvent être séparée de celle de la transformation F . C'est le cas du déplacement rigide, de la similitude, de l'étirement et de l'affinité. Alors, pour ces transformations, on réécrit la fonctionnelle (2) en introduisant G , défini par $F(X) = G(X) + T$, où T est la translation :

$$e(G, T) = \sum_{i=1}^n \|G(X_i) + T - X'_i\|^2 \quad (3)$$

Pour les transformations qui nous intéressent, G est linéaire, en conséquence, on déduit par dérivation par rapport à T que :

$$T = \frac{1}{n} \sum_{i=1}^n X'_i - \frac{1}{n} \sum_{i=1}^n G(X_i) = \overline{X}'_i - G\overline{X}_i \quad (4)$$

Ainsi, il est avantageux de travailler dans les repères centrés sur les barycentres de chaque ensemble, soit $Y'_i = X'_i - \overline{X}'_i$ et $Y_i = X_i - \overline{X}_i$. Nous noterons donc par Y les coordonnées dans le repère centré. Dans ce dernier, le problème de minimisation s'écrit alors simplement, puisque la translation n'apparaît plus :

$$e(G) = \sum_{i=1}^n \|G(Y_i) - Y'_i\|^2 \quad (5)$$

Par là-même, le problème de moindres carrés de départ (3) se réduit à résoudre les deux étapes suivantes :

- trouver $G()$ qui minimise e dans (5),
- puis, calculer la translation avec (4),

Ce schéma de calcul est appliqué dans chacune des quatre sections qui suivent.

Déplacement rigide/Rotation Dans cette section, $G()$ est une rotation de matrice R . Nous présentons l'algorithme pour résoudre le problème de moindres carrés, dans le cas d'une rotation, par la méthode de la décomposition en valeurs singulières (SVD: Singular Value Decomposition). Nous avons préféré présenter celui-ci plutôt que ceux fondés sur les quaternions, car il est utilisable dans un espace de dimension quelconque. Pour avoir la démonstration du résultat et de ses propriétés, il faut se reporter à [30]. Néanmoins, en pratique, dans les cas particuliers de la dimension 2 et 3, nous utilisons l'algorithme avec les complexes et les quaternions [18] respectivement. L'algorithme avec SVD se résume de la façon suivante :

- Calculer la matrice de covariance croisée des deux ensembles de points, dans les repères centrés sur les barycentres $Q = \frac{1}{n} \sum_{i=1}^n Y_i' Y_i^t = \bar{Y}_i' \bar{Y}_i^t$ où l'exposant t indique la transposition.
- Trouver la décomposition en valeurs singulière de Q , $Q = Z D' W^t$, où Z et W sont des rotations et D' est diagonale.
- La meilleure rotation est alors $R = Z W^t$.
- La meilleure translation est $T = \bar{X}_i' - R \bar{X}_i$

À condition que l'ensemble $\{X_i\}_{1,\dots,n}$ ne soit pas dégénéré, la solution au problème est unique. Un ensemble est dégénéré, si ses points sont coplanaires, ou inclus dans un sous-espace de \mathbb{R}^p . Dès que l'on ne peut plus ordonner strictement la diagonale de D' , la décomposition en valeurs singulières n'est plus unique. Pourtant, la rotation optimale résultante est identique pour toutes les décompositions de Q .

L'algorithme d'obtention d'un déplacement optimal sert de base aux méthodes de recalage rigide 2D et 3D. Il est en particulier utilisé dans l'algorithme ICP [5].

Similitude/Rotation et homothétie Pour une transformation de type similitude, G est le produit d'un scalaire positif s , avec une rotation R . Puisque $G(Y) = sRY$, la fonction à minimiser est :

$$e(s, R) = \sum_{i=1}^n \|sRY_i - Y_i'\|^2 \quad (6)$$

La dérivée partielle de $e(s, R)$ par rapport à s est nécessairement nulle, pour minimiser cette équation. Cette contrainte permet d'obtenir s en fonction de la rotation :

$$s = \frac{\sum_{i=1}^n Y_i'^t RY_i}{\sum_{i=1}^n Y_i^t Y_i} \quad (7)$$

Cette formule se réécrit en introduisant la matrice de covariance croisée $Q = \frac{1}{n} \sum_{i=1}^n Y_i' Y_i^t$ et la matrice d'inertie du première ensemble $I = \sum_{i=1}^n Y_i Y_i^t$, de la façon suivante :

$$s = \frac{\text{trace}(RQ^t)}{\text{trace}(I)} \quad (8)$$

Puisque s n'est qu'un facteur, on peut appliquer la décomposition en valeurs singulières utilisée dans la section précédente sans modifications. En effet, les rotations Z et W ne sont pas affectées par ce scalaire s , contrairement à D' . La rotation R est donc égale à ZW^t , si $Q = ZD'W^t$. Par substitution, dans (8), on vérifie que la valeur de s obtenue est positive, puisque D' est positive tout comme I . Comme dans la section précédente, l'unicité de la solution optimale est assurée pour les ensembles de points non dégénérés, s étant positif.

Le résumé de notre algorithme pour les similitudes est donc le suivant :

- Calculer la matrice de covariance croisée centrée $Q = \overline{Y_i' Y_i^t}$ et la matrice de covariance centrée $I = \overline{Y_i Y_i^t}$.
- Trouver la SVD de Q , $Q = ZD'W^t$
- La meilleure rotation est alors $R = ZW^t$.
- Le meilleur facteur d'échelle est $s = \frac{\text{trace}(D')}{\text{trace}(I)}$.
- La meilleure translation est $T = \overline{X_i}' - sR\overline{X_i}$.

Cet algorithme permet de concevoir des méthodes de recalage avec échelle inconnue.

Étirement/Rotation et changement d'échelle Le cas des étirements est plus complexe que le précédent, car il n'a pas de solution générale connue en dimension quelconque. L'ensemble des étirements ne forme en effet pas un groupe de transformations. Néanmoins, nous décrivons la solution du problème dans le cas 2D que nous avons dérivé. En dimension 2, on note θ l'angle de la rotation $R = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$, et $D = \begin{pmatrix} \lambda & 0 \\ 0 & \mu \end{pmatrix}$. Dans le repère des barycentres, la fonction à minimiser est :

$$e(\lambda, \mu, \theta) = \sum_{i=1}^n (\lambda x_i \cos \theta - \mu y_i \sin \theta - x'_i)^2 + (\lambda x_i \sin \theta + \mu y_i \cos \theta - y'_i)^2 \quad (9)$$

où λ et μ sont des scalaires positifs. À partir de la dérivée partielle de $e(\lambda, \mu, \theta)$ par rapport à λ et μ , nous déduisons la valeur des facteurs d'échelle en fonction de l'angle de rotation θ :

$$\lambda = \frac{\overline{x'_i x_i} \cos \theta + \overline{y'_i x_i} \sin \theta}{\overline{x_i^2}} \quad (10)$$

$$\mu = \frac{\overline{y'_i y_i} \cos \theta - \overline{x'_i y_i} \sin \theta}{\overline{y_i^2}} \quad (11)$$

Par substitution de λ et μ dans la dérivée partielle de $e(\lambda, \mu, \theta)$ par rapport à θ , il vient que θ doit vérifier l'égalité suivante :

$$b \cos 2\theta - a \sin 2\theta = 0$$

avec $a = \frac{1}{2} \left(\frac{\overline{x'_i x_i}}{\overline{x_i^2}} - \frac{\overline{y'_i x_i}}{\overline{x_i^2}} - \frac{\overline{x'_i y_i}}{\overline{y_i^2}} + \frac{\overline{y'_i y_i}}{\overline{y_i^2}} \right)$ et $b = -\frac{\overline{x'_i x_i} \overline{y'_i x_i}}{\overline{x_i^2}} + \frac{\overline{x'_i y_i} \overline{y'_i y_i}}{\overline{y_i^2}}$. En conséquence, nous proposons l'algorithme suivant pour les étirements 2D :

- Calculer la matrice de covariance croisée centrée : $Q = \begin{pmatrix} \overline{x'_i x_i} & \overline{x'_i y_i} \\ \overline{y'_i x_i} & \overline{y'_i y_i} \end{pmatrix}$.
- Le meilleur angle de rotation est $\frac{1}{2} \arctan \frac{b}{a} + k\frac{\pi}{2}$. k est déterminé du fait que λ et μ doivent être positifs dans l'étape suivante.
- Les meilleurs facteurs d'échelle sont donnés par (10) et (11).
- La meilleure translation est $T = \overline{X}'_i - \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} \lambda & 0 \\ 0 & \mu \end{pmatrix} \overline{X}_i$.

L'existence et l'unicité de la solution optimale est assurée dès que trois points au moins sont non alignés.

Ce type de transformation, bien que peu utilisé, a pourtant des applications intéressantes.

Affinité/Transformation linéaire Une affinité, après séparation de sa partie translation, est une transformation linéaire, $G(Y) = LY$ avec L une matrice quelconque. Alors, la solution est très simple à obtenir. La solution optimale peut être écrite de façon directe par :

$$L = \overline{Y_i' Y_i^t} \overline{Y_i Y_i^t}^{-1} \quad (12)$$

Il est remarquable de voir, une nouvelle fois, apparaître la matrice de covariance croisée $Q = \overline{Y_i' Y_i^t}$ qui joue un rôle central dans ces problèmes d'estimation de transformations avec correspondance. Pour qu'une solution existe, la matrice de covariance centrée $I = \overline{Y_i Y_i^t}$ doit être inversible. Ainsi, pour les ensembles de points $\{X_i\}_{1,\dots,n}$ non dégénérés, la solution au problème de moindres carrés affine et linéaire existe toujours et elle est unique.

En introduisant la représentation homogène $\mathcal{X} = \begin{pmatrix} X \\ 1 \end{pmatrix}$ d'un point X et la représentation homogène $\mathcal{A} = \begin{pmatrix} L & T \\ 0 & 1 \end{pmatrix}$ d'une transformation affine, l'algorithme peut se résumer à :

- Calculer la matrice de covariance croisée en coordonnées homogènes $Q = \overline{\mathcal{X}_i' \mathcal{X}_i^t}$ et la matrice de covariance en coordonnées homogènes $\mathcal{I} = \overline{\mathcal{X}_i \mathcal{X}_i^t}$.
- La meilleure transformation est $\mathcal{A} = Q\mathcal{I}^{-1}$.

Cet algorithme de régression linéaire sur-déterminé est très classique et très souvent utilisé en vision par ordinateur.

Homographie Les coordonnées homogènes introduites dans la section précédente permettent de représenter une homographie sur un espace de dimension p par une matrice \mathcal{H} de dimension $(p+1) \times (p+1)$. La fonctionnelle (2) qui permet de minimiser l'erreur entre les coordonnées \mathcal{X}_{ij}' de \mathcal{X}_i' et celles de X_i transformées par \mathcal{H} , s'écrit donc :

$$e(\mathcal{H}) = \sum_{i=1}^n \sum_{j=1}^p \left(\frac{H_j \mathcal{X}_i}{H_{p+1} \mathcal{X}_i} - \mathcal{X}_{ij}' \right)^2 \quad (13)$$

où H_1, H_2, \dots, H_{p+1} sont les $p+1$ lignes de l'homographie \mathcal{H} . Puisque cette équation n'est pas linéaire en tous les termes de \mathcal{H} , on pose la fonctionnelle linéaire e' (14) dont la minimisation fournit une solution généralement assez proche de la solution optimale de e (13). Dans le cas où la plus grande précision est recherchée, la solution de e' peut servir de point initial pour une méthode de minimisation itérative de e .

$$e'(\mathcal{H}) = \sum_{i=1}^n \sum_{j=1}^p (H_j \mathcal{X}_i - \mathcal{X}'_{ij} H_{p+1} \mathcal{X}_i)^2 \quad (14)$$

L'équation (14) est un problème de minimisation aux moindres carrés, linéaire. Soit \mathcal{H}^* le vecteur constitué des termes de la matrice \mathcal{H} . Ce vecteur \mathcal{H}^* est supposé de norme 1. Cette condition est ajoutée pour que le problème soit bien posé et que la solution soit unique. Avec les notations précédentes, l'équation (14) se réécrit comme un problème de minimisation sous la contrainte $\|\mathcal{H}^*\| = 1$ de :

$$e'(\mathcal{H}^*) = \sum_{i=1}^{np} (J_i \mathcal{H}^*)^2 \quad (15)$$

Dans cette formule \mathcal{H}^* est un vecteur à $(p+1)^2$ éléments, et J est une matrice $(p+1)^2 \times np$. La solution optimale est donc le vecteur propre de la matrice $J^t J$ associé à la plus petite valeur propre.

Comme dans le cas affine, la solution existe seulement si l'ensemble $\{X_i\}_{1,\dots,n}$ n'est pas dégénéré, c'est-à-dire s'il n'est pas inclus dans un sous-espace de \mathbf{R}^p . On peut montrer que la condition nécessaire et suffisante est réalisée par l'existence d'au moins deux points non coplanaires avec les autres points [9].

Nous pouvons donc résumer l'algorithme dans le cas homographique comme :

- Calculer la matrice J définie par les équations (14) et (15).
- L'homographie optimale est donnée par le vecteur propre associé à la plus petite valeur propre de la matrice $J^t J$.

Ce type d'algorithme est souvent aussi utilisé en auto-calibration, ou lorsque l'on cherche à retrouver l'homographie qui permet à une reconstruction projective de devenir euclidienne [9, 58].

3.1.3 Convergence

L'algorithme GICP converge toujours de façon monotone vers un minimum qui minimise localement le critère aux moindres carrés (1).

La preuve de la convergence de l'algorithme GICP est exactement la même que celle qui permet de démontrer la convergence de l'algorithme ICP, mais elle est appliquée dans un contexte plus général, où la transformation n'est pas seulement une transformation rigide. En effet, la démonstration ne nécessite pas d'avoir une transformation rigide, mais elle réclame que la transformation qui minimise l'équation (1) existe pour toutes les itérations de l'**étape g'**. Or nous pouvons assurer que ces conditions sont vérifiées pour les transformations que nous avons considérées (voir section 3.1.2).

En résumé, les raisons de la convergence de l'algorithme GICP tiennent à ce que l'**étape g** et l'**étape g'** réduisent chacune, à chaque itération, la distance moyenne (1) entre le modèle et les données. L'**étape g**, en choisissant le point le plus proche X'_i de $F(X_i)$, réduit individuellement un terme de la somme (1), et par là-même la distance moyenne. Quant à l'**étape g'**, elle réduit par définition la distance moyenne, puisqu'elle consiste à trouver la transformation qui minimise cette erreur. Alors, comme e_{GICP} est toujours positive, l'algorithme converge de manière monotone vers un minimum.

La figure 2 montre les résultats obtenus après convergence de l'algorithme GICP, sur les mêmes données pour les transformations présentées précédemment. La forme de référence utilisée est le carré unité dans son repère propre.

3.1.4 Points parasites

L'algorithme ICP suppose que les données sont incluses dans la forme modèle. En conséquence, la présence de points parasites dans les données qui ne peuvent pas être recouverts par la forme modèle recalée, biaise les résultats de manière significative.

Il faut donc trouver de façon robuste la partie des données qu'il faut recalculer. Cela nécessite d'être capable de différencier les données potentiellement intéressantes des autres. Il faut donc avoir une caractérisation de ce que sont les données aberrantes. Divers travaux se sont attachés, dans le cadre ICP, à développer des versions robustes aux points parasites, en utilisant différents critères statistiques pour les éliminer [57, 33, 20]

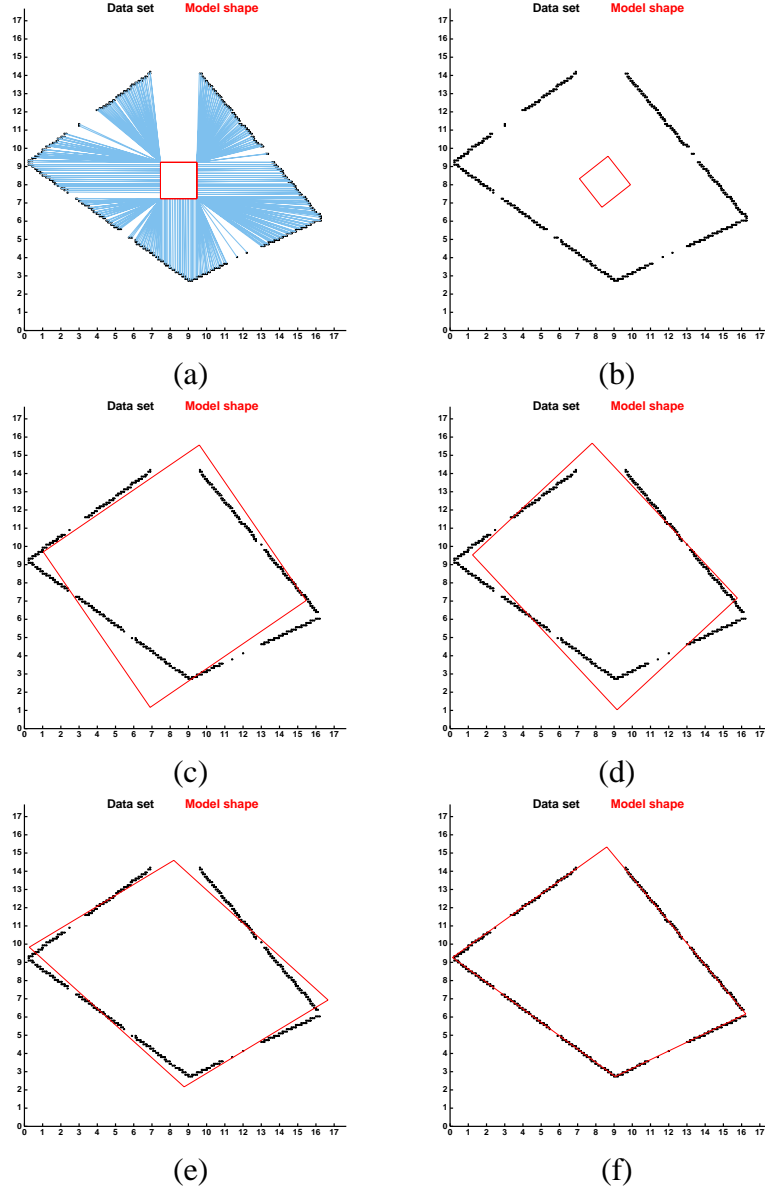


FIG. 2 – Résultat produit par l'algorithme GICP où la forme de référence utilisée est le carré unité dans son repère propre et la transformation (a) une translation, (b) un déplacement rigide, (c) une similitude, (d) un étirement, (e) une affinité et (f) une homographie. On remarque que les données peuvent être tronquées de façon importante sans poser de problèmes à l'algorithme. En (a), les appariements entre la forme de référence et les données sont aussi montrés.

Mais, avec ces méthodes améliorées, comme cela l'a été remarqué [20], moins la transformation est contrainte, plus l'algorithme GICP va pouvoir converger vers une forme dégénérée incluse dans un sous espace. En conséquence, il est nécessaire de contraindre la méthode GICP à expliquer toutes les données, ou alors caractériser de façon plus précise un point aberrant. L'approche floue de la classification permet de faire dans un cadre cohérent.

3.2 Méthodes de classification

L'approche GICP est limitée du fait qu'une seule forme modèle peut être utilisée en même temps. Il s'agit donc d'ajouter le cadre algorithmique de la classification à l'algorithme GICP pour obtenir une méthode de modélisation géométrique multi-modèles.

D'après [44], le cadre de la classification pour faire du recalage est plus satisfaisant que celui fourni par le paradigme hypothèse-vérification pour faire du recalage. Nous pouvons ajouter à ces arguments que le cadre de la classification est intéressant car il permet d'étendre le recalage dans le cas où plusieurs transformations se partagent les données. De plus, la classification a introduit depuis longtemps les méthodes d'estimation robuste par la définition d'une classe de rejet. En particulier, opérer avec deux classes, dont une de rejet, est une méthode d'estimation robuste du type M-estimateur.

Il existe de nombreuses méthodes de classification, mais nous avons choisi l'approche floue car elle est réputée pour ses qualités quant à sa convergence, et comme nous allons le montrer, cette méthode permet de traiter les cas où les classes se recoupent, plus efficacement. Plus profondément, l'approche floue transforme un problème binaire en un problème continu, donc permet de transformer une recherche combinatoire en une estimation par minimisation d'une fonctionnelle.

3.2.1 L'approche floue

La modélisation des problèmes de vision avec des sous-ensembles flous [10] permet de reporter les prises de décision en phase ultime d'un processus de traitement, minimisant ainsi la propagation des erreurs décisionnelles dans des phases intermédiaires. Des approches locales ou globales peuvent être utilisées pour définir les fonctions d'appartenance floues caractéristiques des propriétés des données

traitées en vision. Le premier type d'approche, proposé par Pal et Rosenfeld, généralise les concepts de topologie et de géométrie (connexité, adjacence, compacité) dans le contexte des sous-ensembles flous. Les approches globales sont fondées sur les techniques de classification adaptative floue introduites initialement par Ruspini, Dunn et Bezdek, qui minimisent une famille de fonctionnelles de base : 'Fuzzy c-Means' (FCM) [6]. Le principe de la classification floue est d'autoriser une appartenance partielle ou distribuée de chaque donnée à l'ensemble des classes de l'espace des caractéristiques.

Nous allons présenter deux algorithmes de classification et leurs avantages respectifs : FCM et l'extension en présence de points parasites de Davé [14].

3.2.2 La classification floue

L'algorithme FCM et ses dérivés ont été utilisés avec succès dans de nombreuses applications en reconnaissance des formes et en segmentation d'images. La classification floue est une généralisation des méthodes de partitions où les classes de l'ensemble des n points dans \mathbf{R}^p sont des sous-ensembles non pas classiques mais flous. Alors, chaque point X_i a un degré d'appartenance u_{ik} , à la classe k , entre 0 et 1. La somme des degrés d'appartenance u_{ik} d'un point sur l'ensemble des classes doit valoir 1. Cette contrainte $\sum_{k=1}^c u_{ik} = 1$ oblige les classes à recouvrir toutes les données par les c sous-ensembles flous.

L'algorithme de classification minimise la fonctionnelle suivante :

$$e_{FCM}(T_k, u_{ik}) = \sum_{i=1}^n \sum_{k=1}^c u_{ik}^m d^2(X_i, T_k) \quad \text{avec} \quad \sum_{k=1}^c u_{ik} = 1 \quad (16)$$

L'algorithme FCM se résume ainsi (la convergence vers un minimum local est démontrée dans [6] quand $m > 1$) :

étape 0 : Donner le nombre de classes c , et le facteur flou m (généralement choisi égal à 1.5). Initialiser aléatoirement la position T_k des centres des classes.

étape f : Générer une nouvelle partition en utilisant l'équation suivante du degré d'appartenance floue :

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d(X_i, T_k)}{d(X_i, T_j)} \right)^{\frac{2}{m-1}}} \quad (17)$$

étape f' : Calculer les nouveaux centres de classes T_k en utilisant l'équation suivante :

$$T_k = \frac{\sum_{i=1}^n u_{ik}^m X_i}{\sum_{i=1}^n u_{ik}^m} \quad (18)$$

étape f'' : Si le centre des classes est stable, il faut s'arrêter, sinon retourner à l'étape f.

C'est le centre de gravité T_k , qui est le prototype de la classe k .

La complexité de chaque étape est de l'ordre de $O(cnp)$, ce qui peut conduire à un coût de calcul important avec un grand nombre de classes.

3.2.3 Avantages de la classification floue

Un des avantages de l'algorithme de classification floue est qu'il utilise une contrainte : la somme des degrés d'appartenance sur les classes est égale à 1. Ainsi, l'algorithme peut traiter les points ambigus, comme le montre la figure 3, ce que ne peut pas faire une méthode de classification classique. En particulier, on observe que les centres des classes sont mieux localisés par la méthode floue en présence de points ambigus. Dans ce chapitre, cette propriété se révèle très utile pour réduire le nombre de minima locaux où la solution est réduite à une forme dégénérée.

Il est aussi possible de traiter les points aberrants grâce à la classification floue en introduisant une classe de rejet comme cela est décrit dans la section suivante.

3.2.4 Classification floue avec des points parasites

Les résultats de l'algorithme précédent peuvent beaucoup changer en présence de points parasites. Davé [14] a étendu le FCM pour prendre en compte ces points

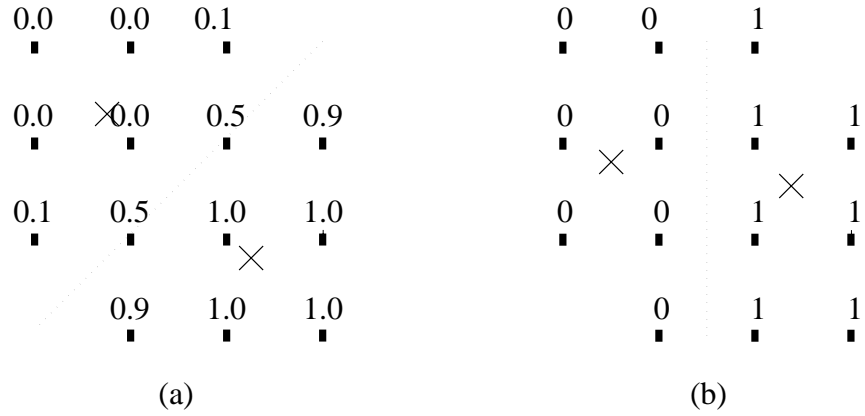


FIG. 3 – Classification obtenue sur les mêmes données avec la méthode de classification floue (a) qui traite mieux les points ambigus, et une méthode classique (b). On remarque en particulier que les centres des deux classes, marqués par des croix, sont mieux localisés par la méthode floue. Le chiffre au dessus point indique son degré d'appartenance à la classe du bas.

aberrants et il a introduit, dans les approches floues, le concept de classe de rejet pour les récolter. Par définition, tous les points sont à la même distance du prototype T_c de la classe de rejet. Pour contrôler cet algorithme, un nouveau paramètre δ est nécessaire, qui permet d'indiquer qu'elle est la distance minimale à partir de laquelle un point est considéré comme trop éloigné du centre d'une classe pour pouvoir lui appartenir.

Une connaissance supplémentaire est donc introduite : le paramètre δ . Ce paramètre spécifie l'épaisseur de la zone d'influence de chaque prototype.

Dans le but d'éviter l'introduction d'une connaissance supplémentaire, Davé a proposé une heuristique pour choisir le paramètre δ . Mais, ici, nous n'avons pas fait ce choix, car ce critère ne nous a pas donné totale satisfaction sur les tests présentés dans la section 3.4 où les classes sont entrelacées. En effet cette heuristique suppose implicitement une bonne séparation des classes.

L'algorithme FCM modifié avec la classe de rejet est le suivant :

étape 0 : Fixer le nombre c de classes, et le facteur flou m (généralement $m = 1.5$).
Calculer la position initiale des centres des classes T_k en utilisant la méthode

FCM. Spécifier la distance δ qui définit la zone d'influence géométrique de chaque prototype.

étape f : Générer une nouvelle partition en utilisant l'équation (17) du degré d'appartenance flou u_{ik} .

étape f' : Calculer le nouveau centre des classes T_k en utilisant l'équation (18) seulement pour k entre 1 et $c - 1$.

étape f'' : Si le centre des classes est stable, il faut s'arrêter, sinon retourner à l'étape f.

Dans cette description, X_i est un des n éléments de l'ensemble des données contenues dans \mathbb{R}^p . $d(X_i, T_k)$ est défini comme la distance euclidienne entre le point X_i et le centre de la classe T_k pour k entre 1 et $c - 1$, et pour la classe de rejet, la distance est $d(X_i, T_c) = \delta$. La valeur de δ doit être choisie avec attention (voir un exemple dans la section 5.2).

La convergence de cet algorithme est assurée. La classe de rejet permet d'avoir une méthode de classification qui intègre un estimateur robuste de type M-estimateur. Un M-estimateur limite l'influence des points aberrants en les pondérant itérativement par une valeur qui décroît avec la distance. La classification fait exactement de même par le biais du degré d'appartenance u_{ik} qui est une fonction inverse de la distance du point X_i au prototype T_k relativement à tous les prototypes (voir équation (17)).

Dans la méthode FCM et son extension, le prototype d'une classe est le centre de gravité. Il paraît très intéressant en vision par ordinateur comme en reconnaissance des formes, de disposer d'un prototype plus générique ayant une forme géométrique, soit ligne, triangle, rectangle, ou ellipse. Des dérivés de FCM existent où la définition du prototype est modifiée pour qu'il soit plus riche géométriquement qu'un point. On trouve des méthodes qui utilisent des lignes et des hyper-plans [34], des hyper-sphères [35], et des hyper-ellipsoïdes [15]. Néanmoins, ces méthodes sont spécifiques à une forme donnée. En utilisant, l'approche ICP, il est possible de dépasser cette limitation et nous proposons dans la section suivante un algorithme générique pour classer des formes géométriques d'un même type, qui peut être utilisé avec un grand nombre de formes, une fois conçue la fonction de plus proche voisin particulière adaptée à chaque forme de prototype.

3.3 Unification des algorithmes GICP et FCM

Il est possible de faire plusieurs rapprochements entre les algorithmes GICP et FCM. D'abord, les deux méthodes minimisent itérativement en deux étapes une fonction de coût aux moindres carrés (1) ou (16) entre un prototype et des données. Par exemple, on voit que l'**étape f'** et l'**étape g'** consistent à faire le même calcul : calculer la transformation géométrique qui minimise l'écart entre la forme prototype ou modèle et les données. Plus particulièrement, le calcul est exactement identique dans ces deux étapes quand le GICP est utilisé avec une translation et une forme modèle qui est réduite à un point, et quand le FCM a une unique classe. Il semble donc naturel d'unifier ces deux approches pour obtenir un algorithme puissant de Modélisation Multi-Prototypes (MMP) dont l'algorithme GICP et FCM ne sont que des cas particuliers. Ceci est réalisable très simplement en ajoutant au GICP le cadre des ensembles flous du FCM.

3.3.1 Description de l'algorithme MMP

L'algorithme de Modélisation Multi-Prototypes sans point parasite (*respectivement avec une classe de rejet F_c*) consiste en :

étape 0 : Fixer le nombre de classes c et la forme du prototype S_{ref} . Initialiser les transformations F_k (voir la section 3.3.3). (*Donner la distance δ*).

étape m : Calculer le plus proche voisin X'_{ik} appartenant au prototype S_{ref} de chaque point $F_k(X_i)$. (*seulement pour k entre 1 et $c - 1$*).

étape m' : Générer une nouvelle partition en utilisant l'équation du degré d'appartenance flou suivante :

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d(F_k(X_i), X'_{ik})}{d(F_j(X_i), X'_{ij})} \right)^{\frac{2}{m-1}}} \quad (19)$$

(où $d(F_c(X_i), X'_{ic}) = \delta$).

étape m'' : Calculer les nouvelles transformations F_k qui minimisent l'erreur pondérée aux moindres carrés entre les points $F_k(X_i)$ et X'_{ik} :

$$\sum_{i=1}^n (u_{ik})^m d^2(F_k(X_i), X'_{ik})$$

(seulement pour k entre 1 et $c - 1$).

étape m''' : Si la partition est stable, il faut s'arrêter, sinon retourner à l'étape m .

L'algorithme MMP minimise la fonction de coût suivante :

$$e_{MMP}(F_k, X'_{ik}, u_{ik}) = \sum_{k=1}^c \sum_{i=1}^n (u_{ik})^m d^2(F_k(X_i), X'_{ik}) \quad \text{avec} \quad \sum_{k=1}^c u_{ik} = 1 \quad \text{et} \quad X'_{ik} \in S_{ref} \quad (20)$$

L'algorithme GICP n'est qu'un cas particulier de l'algorithme MMP, où le nombre de classe c est égal à 1 (on a donc $u_{i1} = 1$). On voit aussi que l'algorithme FCM est un autre cas particulier, où le prototype S_{ref} est réduit à un point et où la transformation F_k est une translation.

3.3.2 Convergence

L'algorithme de modélisation Multi-Prototypes converge toujours de façon monotone vers un minimum local de l'erreur aux moindres carrés (20).

Les grandes lignes de la démonstration sont analogues à celles de l'algorithme GICP et FCM. Le principe est que chaque étape réduit le critère de l'écart aux moindres carrés (20). Certes, cette convergence est uniquement vers un minimum local. Alors, comment initialiser les prototypes pour obtenir une solution intéressante?

3.3.3 Initialisation

Avec l'algorithme itératif ICP, la position de départ est souvent supposée connue. En effet, la solution que l'on obtient dépend de manière critique de la position initiale. L'algorithme FCM, ne prenant pas en compte les points parasites, est plus robuste à l'initialisation qui peut souvent alors être aléatoire.

L'algorithme MMP est, lui aussi, assez sensible à l'initialisation. Nous proposons donc le schéma suivant pour initialiser des transformations F_k :

- Trouver la partie de F_k qui est une translation T_k par identification de la translation avec le centre des classes qui sont solutions de l'algorithme FCM.

- Utiliser les matrices d'inertie des sous-ensembles flous obtenus précédemment pour calculer, par diagonalisation, la rotation initiale, ou l'étirement initial de chaque prototype, suivant le besoin.

Naturellement, ce schéma est aussi étendu pour former un schéma d'initialisation du plus approximatif au plus fin : on trouve la meilleure translation, le meilleur déplacement, le meilleur étirement, la meilleure affinité, et enfin la meilleure homographie (voir la figure 2). Ces étapes peuvent mélanger, suivant le problème, la version avec ou sans la classe de rejet.

Ces schémas nous ont donné satisfaction dans certains de nos tests (voir la figure 5(c) où les translations apparaissent comme des croix). Mais, comme cela est indiqué dans la section suivante, certains cas particuliers, où les formes sont très fortement entrelacées, mettent en échec ce schéma d'initialisation. Il faut alors, pour chaque type de configurations à problème, définir une façon adaptée d'initialiser l'algorithme MMP. Trouver une initialisation astucieuse, mais spécifique à un type de données, est aussi utile pour accélérer l'algorithme.

3.4 Résultats

3.4.1 Convergence rapide dans les premières étapes

Dans la figure 4, les points échantillonnent approximativement et partiellement un cercle et un parallélogramme. Ces points sont modélisés comme 2 parallélogrammes (figure 14(a)).

La figure 4, où est visible une itération toutes les 10 itérations, donne une idée de la recherche effectuée par l'algorithme. Le nombre d'itérations nécessaires pour arriver au résultat est de 156 (le résidu e_{MMP} avait alors une variation inférieure 0.001 à chaque étape).

Quant aux prototypes initiaux, ils sont obtenus par le schéma d'initialisation décrit dans la section 3.3.3. Cette figure présente un cas relativement difficile pour l'algorithme car l'initialisation est assez éloignée de la solution à trouver, puisque les deux formes sont entrelacées.

Mais, nous avons observé, en général, une convergence très rapide dans les premières étapes lorsque l'initialisation est proche de la solution recherchée. Les étapes supplémentaires de stabilisation peuvent en revanche être assez nombreuses (de l'ordre d'une centaine). Néanmoins, le nombre d'étapes ne peut être tronqué

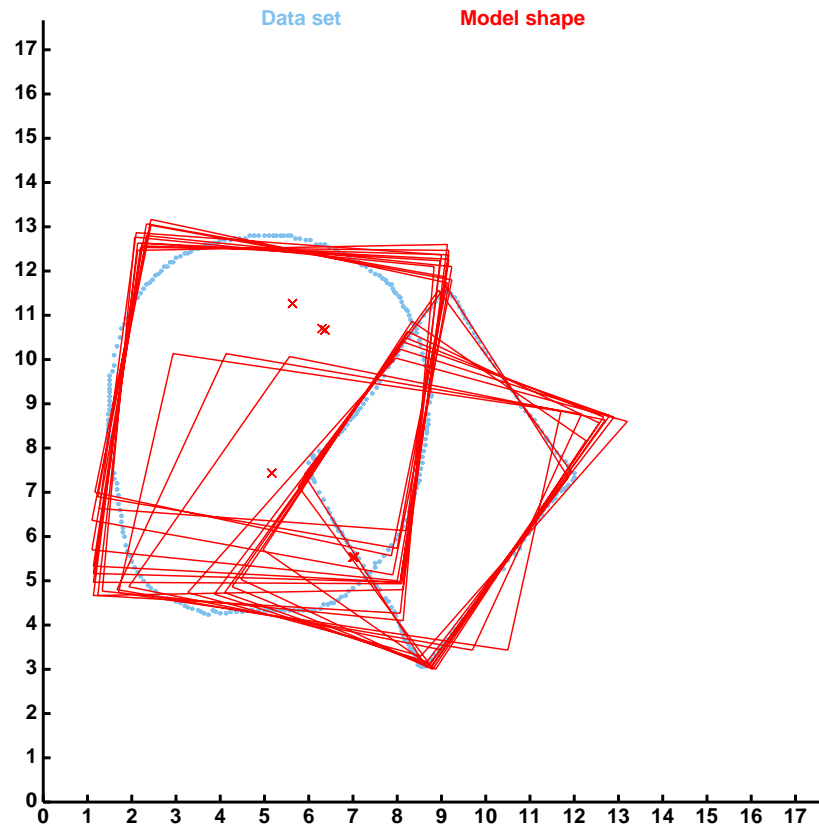


FIG. 4 – Quelques itérations de notre algorithme. Le résultat final est visible sur la figure 14(a).

trop brutalement si on désire conserver la robustesse de l'algorithme quand la solution initiale est assez éloignée de la transformation optimale.

3.4.2 Données entrelacées

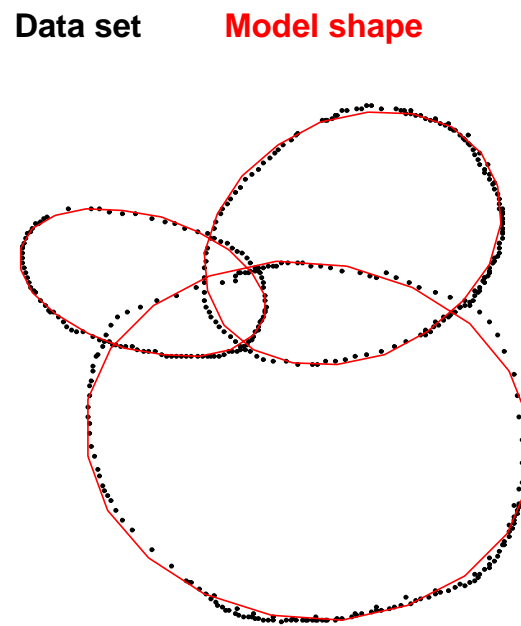


FIG. 5 – L'algorithme MMP interprète bien l'ensemble comme trois ellipses qui s'intersectent.

Dans la figure 5, l'ensemble des points est synthétique, bruité et incomplet. Il est modélisé avec justesse par l'algorithme MMP comme trois ellipses qui s'inter-

sectent. Les prototypes initiaux sont obtenus par le schéma d'initialisation décrit dans la section 3.3.3.

Néanmoins, dans des configurations de ce type, où les objets à modéliser sont fortement entrelacés, le schéma d'initialisation de l'algorithme MMP ne donne généralement pas de bonnes formes de départ. En effet, notre schéma d'initialisation suppose que les parties à recaler par chaque prototype géométrique sont bien séparées. De façon analogue, lorsque les ellipses sont concentriques, le schéma d'initialisation proposé est mis en échec car la première étape de l'initialisation ne permet pas d'obtenir une classification pertinente de l'ensemble. Mais, dans certains cas particuliers, un schéma de l'algorithme MMP adapté peut être proposé. Par exemple, pour les ellipses concentriques, on peut supposer les centres tous identiques et par une classification unidimensionnelle de l'histogramme des distances au centre, trouver les petits et grands axes initiaux.

3.4.3 En présence de points parasites

La figure 6 montre un exemple 2D en présence de 20% de points aberrants. Dans ce cas précis, trois carrés sont présents, et sont retrouvés par notre algorithme en procédant en deux étapes :

- un recalage rigide avec 3 carrés unités sans introduire de classe bruit,
- un recalage par similitude des 3 carrés obtenus par l'étape précédente avec en plus une classe pour récolter les points parasites (δ distance des points parasites a été choisie égale à 20% du côté des cubes).

Lorsque les points parasites sont uniformément répartis, comme dans la figure 6, initialiser l'algorithme MMP par une recherche avec un déplacement rigide ou une similitude, sans classe parasite est souvent suffisant. En effet, les points aberrants n'induisent alors pas de perturbation dans une direction privilégiée. Par contre, en présence d'un point d'accumulation de points parasites, la procédure précédente échoue. Il faut alors mieux considérer le point d'accumulation comme une nouvelle structure dégénérée, et faire une recherche avec une classe supplémentaire comme cela a été fait dans la figure 7.

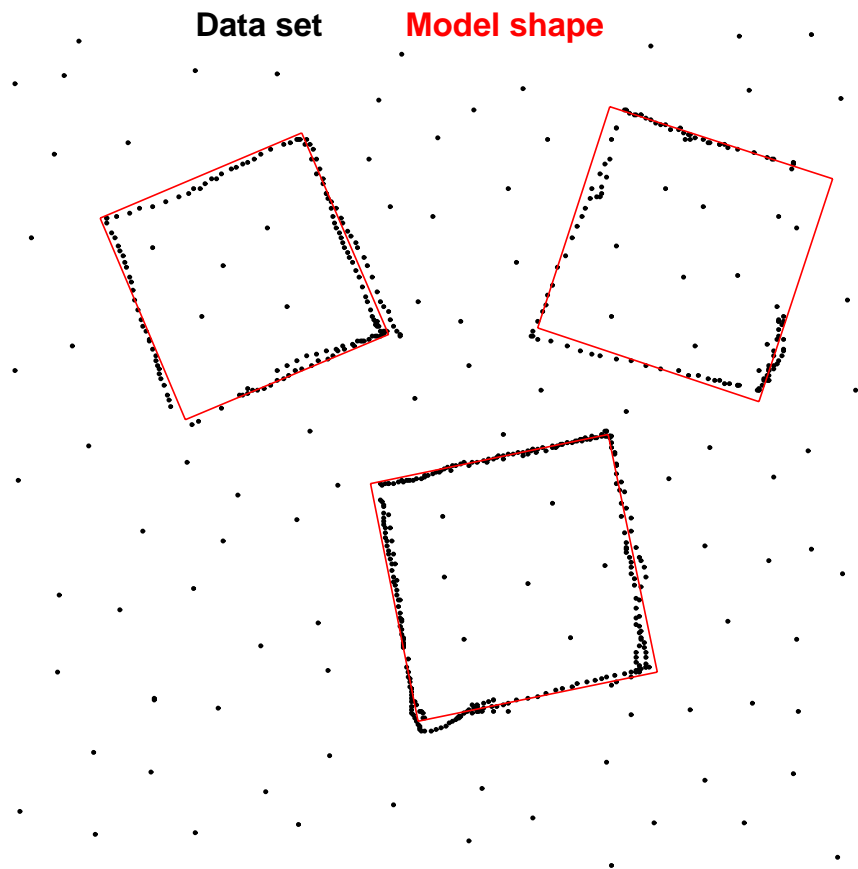


FIG. 6 – L'algorithme MMP interprète bien l'ensemble comme trois cubes malgré le bruit de fond représentant environ 20% des données.

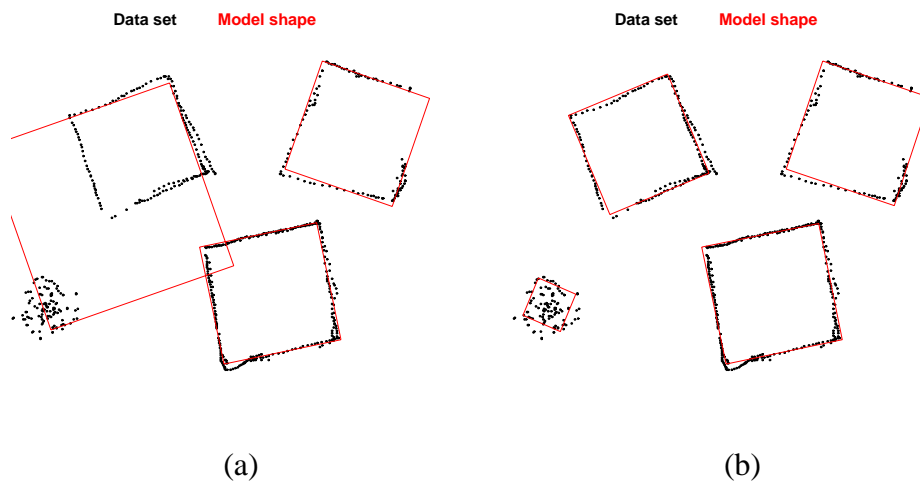


FIG. 7 – En (b), l'algorithme MMP interprète bien les trois cubes malgré car le point d'accumulation de points parasites est considéré comme une primitive dégénérée. Alors que si l'on utilise la version robuste, l'algorithme n'est pas capable de séparer les points parasites qui représentent 20% des données.

3.4.4 Sur des reconstructions 3D

Suivant le choix du type prototype, il est intéressant de modéliser des objets de forme polyédrique ou gauche.

Avec le système de modélisation décrit dans ce document, la modélisation par plusieurs parallélépipèdes est plus particulièrement intéressante. En effet, beaucoup d'objets de notre environnement sont composés d'un assemblage de ces formes, les bâtiments en particuliers.

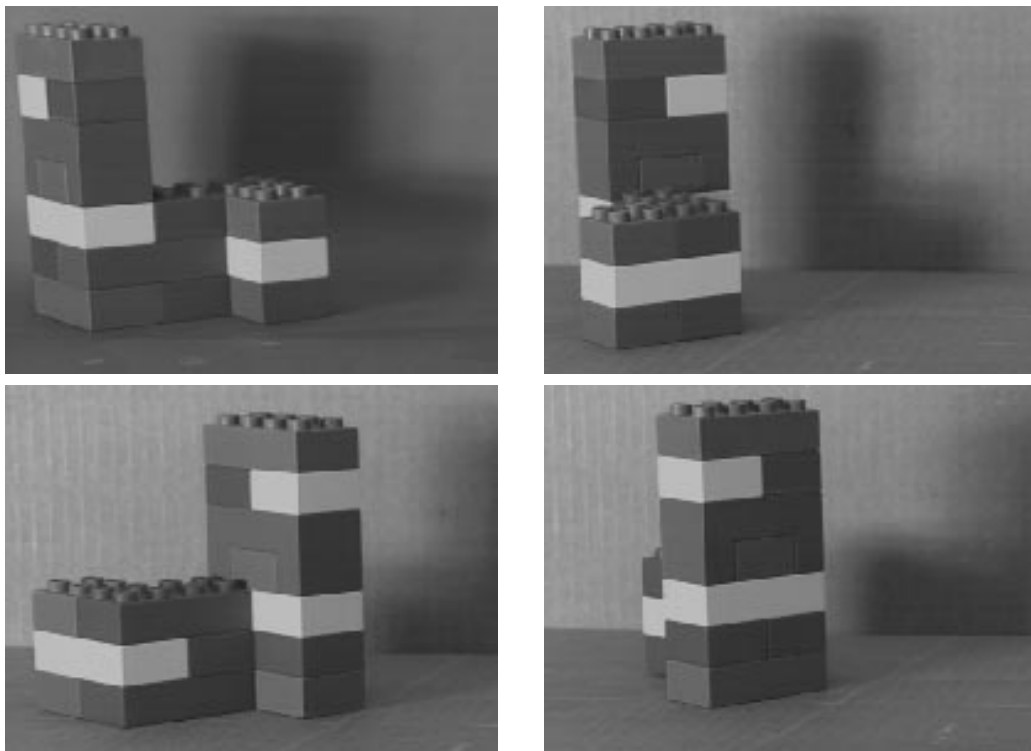


FIG. 8 – Images gauches des quatre paires stéréoscopiques originales d'un jeu de Lego.

Pour illustrer notre procédure de modélisation 3D à partir de paires stéréoscopiques, nous avons, à titre d'exemple, réalisé la modélisation complète d'un objet très simple constitué de pièces d'un jeu de Lego. Cet objet, visible sur la figure 8,

est constitué de deux blocs principaux interconnectés. Nous avons pris quatre paires stéréoscopiques de cet objet selon les points de vue, Sud, Est, Nord et Ouest.

Une fois la calibration réalisée par la méthode décrite dans le deuxième chapitre, les images sont rectifiées. Puis les images rectifiées ont été segmentées par une méthode par ‘split and merge’ [43] en utilisant au mieux l’information de couleur de chaque région. Les régions obtenues dans chaque image sont appariées selon des critères de compatibilité épipolaire et de couleurs et tailles proches [52].

Alors, avec la méthode géométrique décrite dans [46], nous avons obtenu les reconstructions selon les quatre points de vue. Elles sont visibles dans la figure 9. Puisque les paires de sont pas calibrées par rapport à un même repère absolu, il est nécessaire de recalcr les reconstructions entre elles. Dans ce but, nous avons utilisé la décrite dans [47, 48] pour recalcr et ainsi fusionner la reconstruction ‘Sud’ avec celle ‘Est’ et la reconstruction ‘Nord’ avec celle ‘Ouest’. Par contre, pour recalcr les deux reconstructions ainsi produites, il a été nécessaire d’utiliser l’algorithme de recalcr fondé sur l’ICP avec une recherche exhaustive sur un échantillonnage des positions initiales possibles [51]. En effet, du fait des symétries de l’objet, le premier algorithme produit en priorité un recalcr où l’objet est sans épaisseur en plaquant les deux reconstructions l’une contre l’autre. La reconstruction complète est visible dans la figure 10.

On remarque combien cette reconstruction présente des défauts et des facettes redondantes. Il est donc nécessaire pour obtenir un modèle plus synthétique de l’ensemble de pièces de Lego, de chercher quelles sont les caractéristiques des primitives qui recouvrent cette reconstruction. Pour réaliser cela, nous avons interpolé les faces pour qu’elles présentent des points uniformément répartis en leur intérieur, puis nous avons utilisé l’algorithme MMP en procédant de manière à retrouver tout d’abord les deux blocs principaux. L’algorithme avec 2 parallélépipèdes a donc été directement appliqué, puis une fois la convergence atteinte, nous avons ajouté une classe de bruit pour affiner le recalcr sur les deux blocs principaux. Cette dernière étape permet aussi de récolter les points qui relient les deux grands blocs car ils se retrouvent dans la classe bruit. Alors, à partir de cette segmentation de la reconstruction en trois classes, nous avons pu initialiser l’algorithme avec trois parallélépipèdes, sans classe de rejet.

Cette façon de procéder, si elle semble un peu compliquée est générique, car elle permet de séparer récursivement les formes principales bien recouvertes par les primitives, des détails qui vont être recouverts à leur tour précisément en introduisant

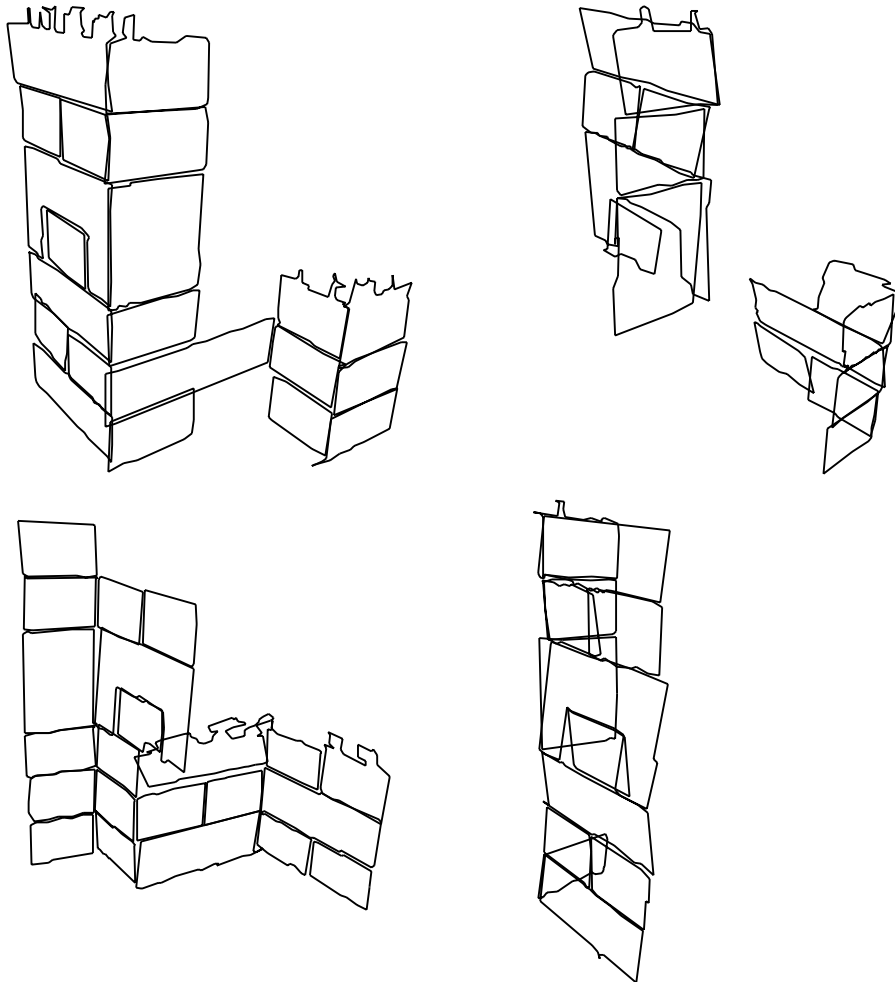


FIG. 9 – Les reconstructions du jeu de Lego obtenues à partir des points de vue de la figure 8. Les vues ont été choisies avec des orientations telles que les reconstructions puissent s’assembler pour constituer visuellement la reconstruction complète.

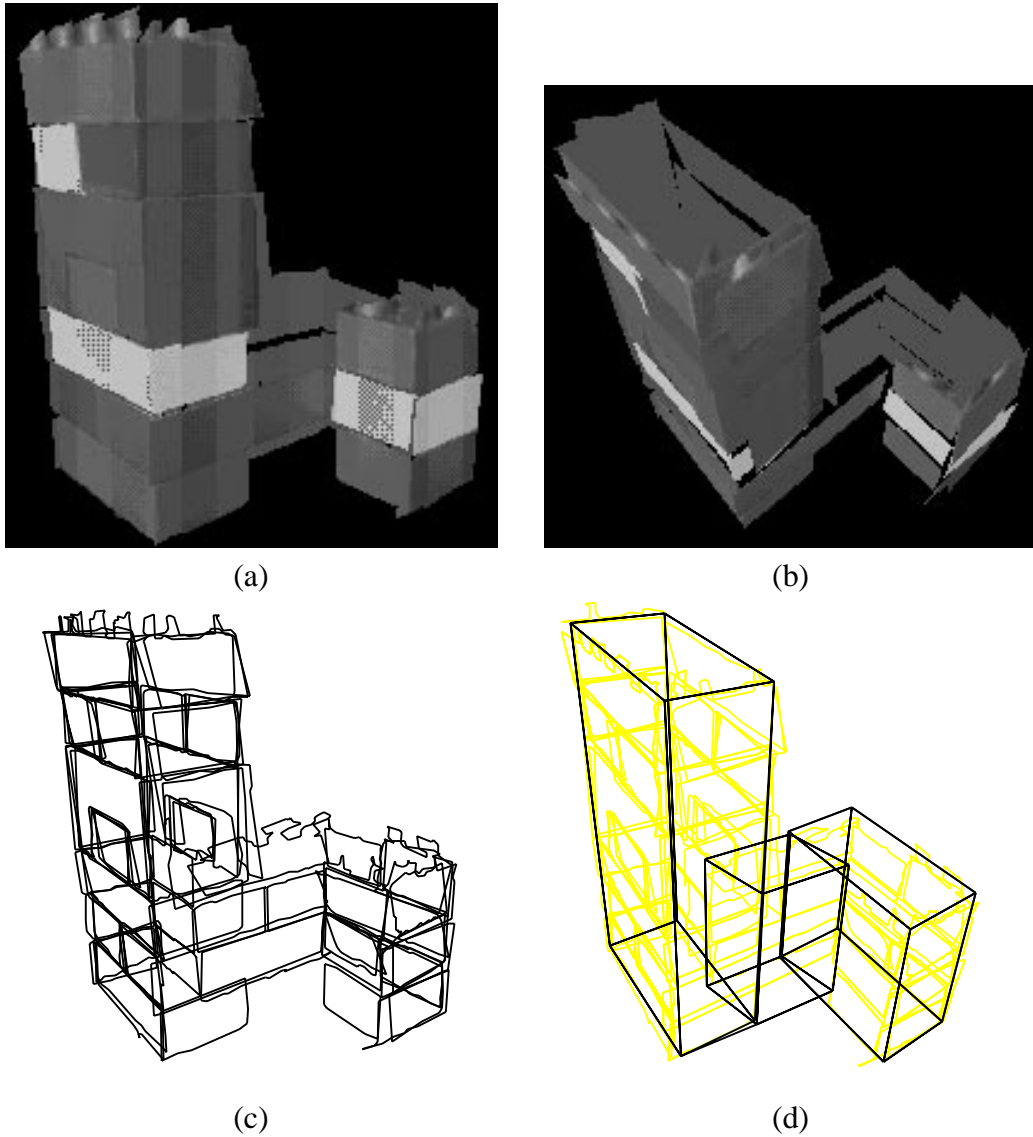


FIG. 10 – (a) et (b) sont deux vues de la reconstruction complète obtenue en recalant les reconstructions de la figure 9. La texture de la première image originale est retroprojetée sur cette reconstruction. (c) est une vue en fil de fer de cette même reconstruction. (d) montre les 3 parallélépipèdes qui recouvrent la reconstruction.

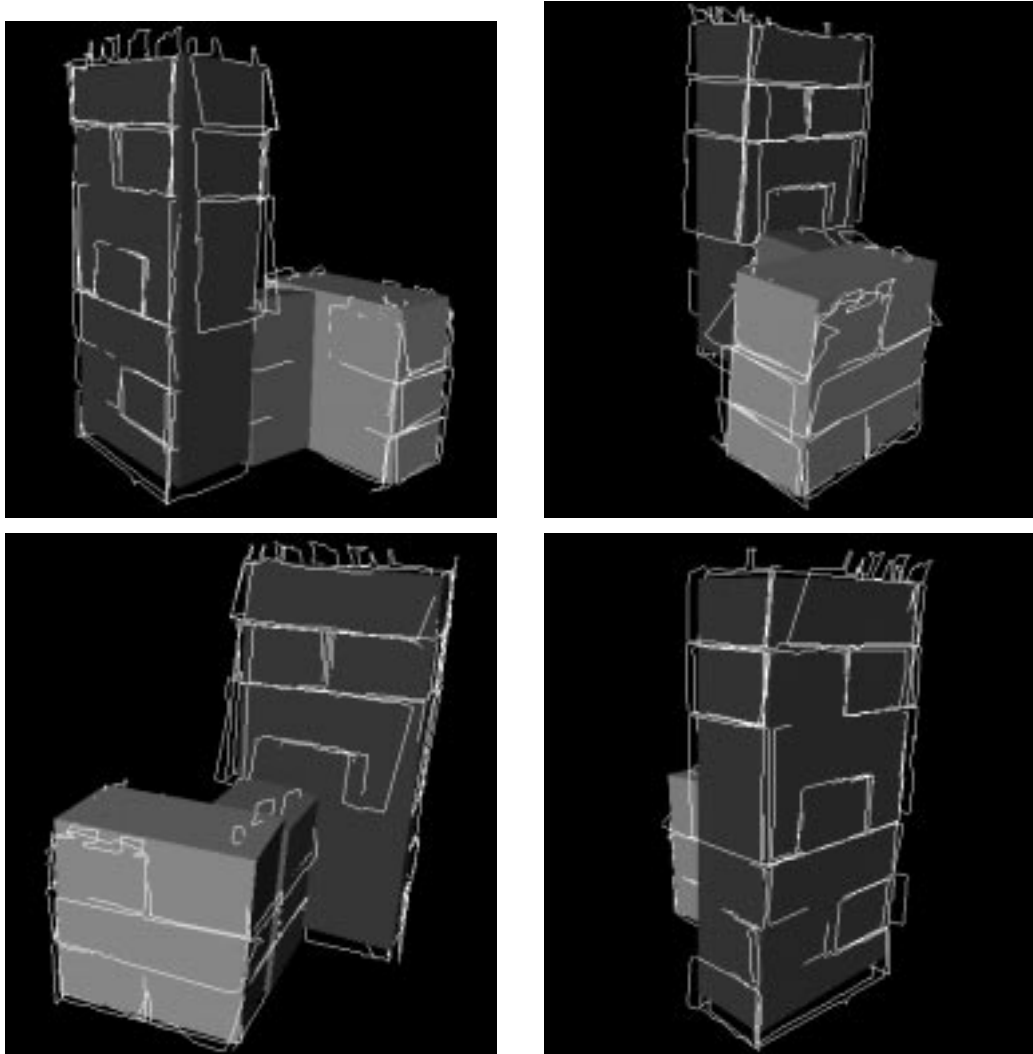


FIG. 11 – Les trois parallélépipèdes sont recalés avec la reconstruction complète du jeu de Lego, et sont vus approximativement selon les quatre lieux d'acquisition des images originales. La taille des primitives est légèrement réduite pour permettre une meilleure visualisation des contours reconstruits.

une nouvelle classe. Cette méthode trouve dans un même temps le recalage à des échelles différentes et détermine automatiquement le nombre de classes utiles. Le résultat, superposé aux contours de la reconstruction est visible dans la figure 11.

Enfin, nous présentons dans la figure 12, le modèle résultat obtenu des pièces de Lego, sur lequel est partiellement retroprojetée la première image originale afin de faciliter la compréhension des vues.

3.4.5 Sur des reconstructions 3D avec normales

Notre algorithme peut aussi s'appliquer à des objets non polyédriques, mais courbes. La figure 13 montre le résultat de l'algorithme MMP quand le prototype est la sphère unité centrée. Les transformations F_k sont affines, et les données, gracieusement fournies par Pascal Fua, sont un modèle 3D de la surface d'une tête.

Pour ces données, comme on disposait en plus des normales à la surface $\{N_i\}_{1,\dots,n}$ en chaque point de l'ensemble $\{X_i\}_{1,\dots,n}$. Nous avons intégré cette information dans l'algorithme pour obtenir une méthode plus robuste. La normale sert uniquement à améliorer la mise en correspondance par le biais du calcul du degré d'appartenance u_{ik} . En effet, dans l'étape m' de l'algorithme MMP qui calcule le degré d'appartenance, nous ajoutons, à la distance entre les points X'_{ik} et $F_k(X_i) = L_k X_i + T_k$, la distance entre les deux vecteurs normaux associés. Cette dernière distance doit être pondérée par un facteur μ , à régler, pour tenir compte de la différence de nature entre un point et un vecteur normé. Plus ce facteur est élevé, plus on tient compte des normales.

Calculer cette distance nécessite de connaître les deux vecteurs normaux. Premièrement, la normale N_i est transformée en $L_k^{-1t}(N_i)$ par une l'affinité. Cette propriété provient du fait que la normale est définie comme orthogonale au plan tangent qui lui se transforme selon F_k . Deuxièmement, il faut calculer la normale à la surface du prototype au point le plus proche X'_{ik} . Comme le prototype est la sphère centrée de rayon unité, cette normale est identique au vecteur X'_{ik} .

Le nouvelle distance est donc :

$$d_{ik} = d(F_k(X_i), X'_{ik}) + \mu d(L_k^{-1t}(N_i), X'_{ik})$$

La figure 13 montre que notre algorithme permet d'obtenir un modèle du crâne de la tête avec des éléments en forme d'ellipsoïdes. Par contre, la faible densité des points et une initialisation brutale n'ont pas permis d'obtenir la modélisation de la face du visage.

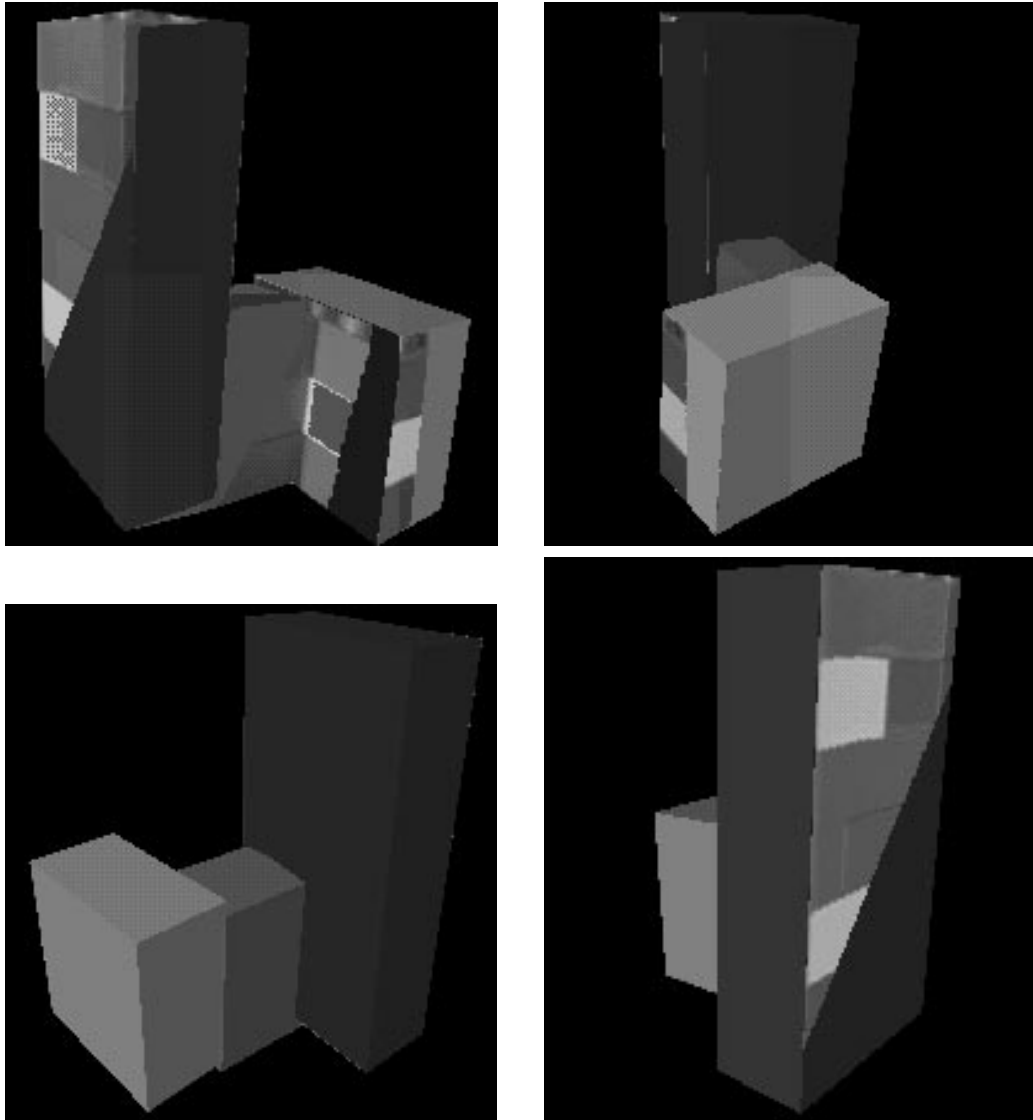


FIG. 12 – Les trois primitives de la figure 11 recalées sur la reconstruction sont vues selon les mêmes positions de prise de vue que celles de la figure 8. De plus, la texture de la première image originale est partiellement retroprojetée sur les primitives afin d'observer la qualité du recalage.

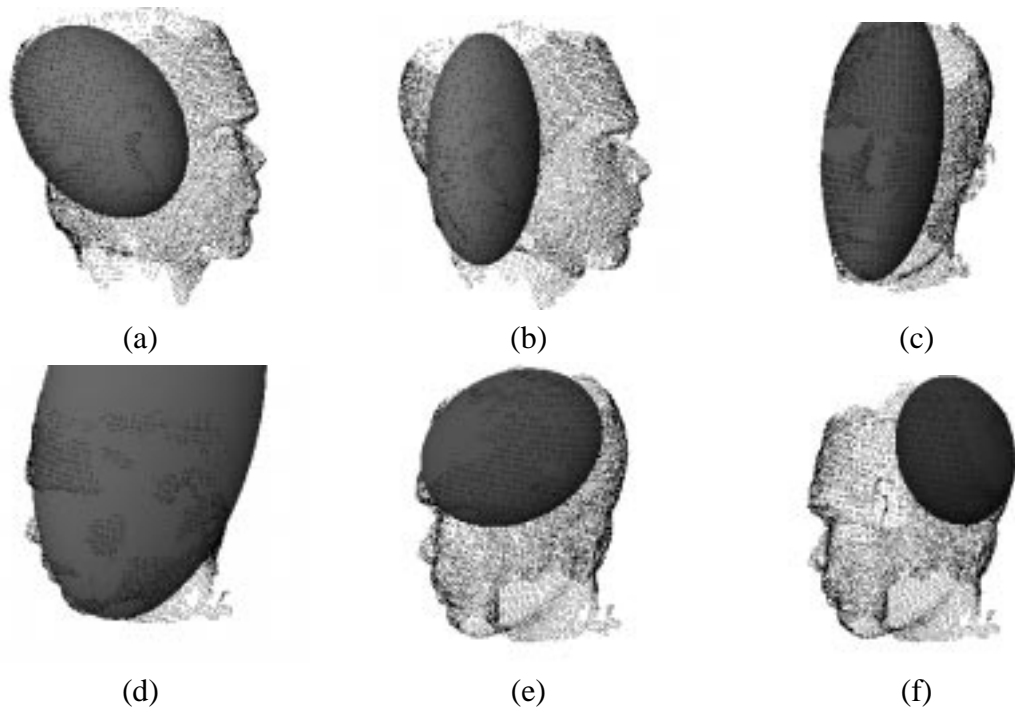


FIG. 13 – Les points et les normales du modèle 3D d'une tête, sont modélisés comme la superposition de 15 ellipsoïdes par l'algorithme MMP (les ellipsoïdes ne sont pas découpées). Ces ellipsoïdes recouvrent de façon satisfaisante le haut (a)(f) et le bas (b) de l'arrière du crâne, le front (d), et les cotés (e)(c) de la tête.

3.4.6 Des prototypes hétérogènes

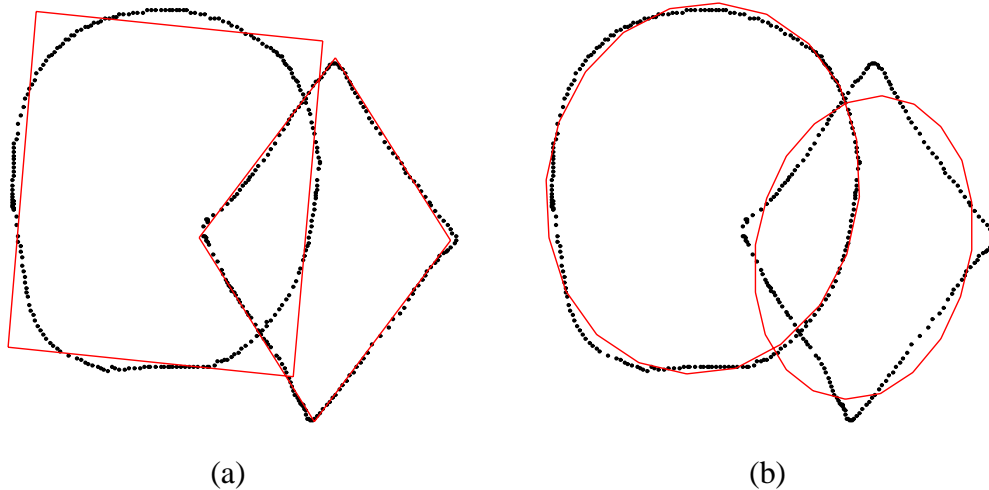


FIG. 14 – *L'ensemble des points est synthétique, mais il est bruité et incomplet. Les points sont sur un cercle et sur un parallélogramme échantillonnés. Cet ensemble est modélisé par l'algorithme MMP comme 2 parallélogrammes en (a), et comme 2 ellipses en (b).*

La figure 14 montre que l'algorithme MMP peut produire des solutions intéressantes même si le prototype ne correspond pas bien aux données. L'algorithme est assez robuste. Cette propriété permet d'envisager l'utilisation de cette méthode pour segmenter les données avec des prototypes hétérogènes comme un parallélogramme et une ellipse, en choisissant la primitive qui minimise au mieux l'erreur moyenne entre les données et le prototype.

Sur l'exemple de la figure 14, le résidu entre l'ovale à gauche est de 0.05 mm avec le prototype carré, alors qu'il n'est que de 0.03 mm avec le prototype cercle. Inversement, le résidu du parallélogramme est de 0.08 mm avec le prototype elliptique, alors qu'il n'est que de 0.03 mm avec le prototype parallélogramme. La discrimination semble donc possible sur cet exemple, bien que les données soient assez déformées et perturbées. Cette technique demande à être testée plus profondément.

3.5 Discussion

Nous avons présenté un algorithme de Modélisation Multi-Prototypes qui permet de recouvrir un ensemble de points par plusieurs instances d'une même primitive paramétrée par une transformation géométrique. Cet algorithme permet de faire de la modélisation par un ensemble de rectangles, de triangles, d'ellipses, d'ellipsoïdes, de parallélépipèdes, ou de toutes autres primitives simples, dès que l'on peut écrire la fonction qui à tout point donne le plus proche voisin sur la primitive.

Une limitation importante de la classification floue, telle que nous l'avons introduite est de donner le nombre c de classes. Plusieurs travaux [34, 54, 56] ont proposé des méthodes pour dépasser cette limitation qu'il serait intéressant d'adapter au cas de notre algorithme MMP.

4 Application à la segmentation des images

Il existe de nombreuses méthodes de segmentation des images [28], par classification de l'histogramme des niveaux de gris ou en couleurs [17], par accroissement de régions [25, 52], par 'split and merge' [43]. Beaucoup de ces méthodes ont trouvé un cadre unificateur par une formalisation variationnelle unifiée [32, 16].

Dans la pratique, ne pas avoir des régions qui sont constantes, mais des régions dont l'intensité varie linéairement est utile [36, 1]. En effet, les gradients dus à l'éclairage peuvent souvent être approximés par des plans. En suivant le même raisonnement, on voit que la segmentation par des régions où l'intensité varie quadratiquement est aussi intéressante comme le montre son utilisation sur les images de profondeur [19], et sur les images d'intensité [7].

Notre algorithme de modélisation MMP, dans sa version sans point aberrant, est aussi une méthode de segmentation de l'ensemble des données, où les points de chaque classe sont contraints à appartenir à un prototype. Notre approche nous a donc permis de développer une méthode de segmentation floue qui étend la classification sur l'histogramme [10] pour permettre la segmentation avec des régions où l'intensité varie linéairement ou quadratiquement.

De plus, grâce à la classification robuste introduite par Davé, nous pouvons faire de la segmentation robuste qui détermine automatiquement les pixels aberrants comme dans [29].

4.1 Algorithme MMP adapté à la segmentation d'images

Nous avons développé une version de l'algorithme MMP spécialement adaptée à la segmentation des images [37]. Une image de profondeur peut être vue comme un nuage de points 3D, par analogie, une image d'intensité aussi. Mais, le quadrillage régulier des images simplifie beaucoup le problème [39], par rapport aux données purement 3D utilisées dans la section précédente. En effet, les données 2D 1/2 permettent de se ramener à une classification 2D et à des estimations géométriques 1D. Alors, chaque prototype est réduit à une fonction de deux variables sur l'image.

Nous avons testé deux versions de l'algorithme, l'une avec des fonctions linéaires, l'autre avec des fonctions quadratiques. Mais, l'ajout d'une simple procédure permet d'avoir un algorithme où les variations seront polynômiales. Il suffit de concevoir la procédure d'estimation adaptée, pour étendre l'algorithme au type désiré de variation de la fonction. Un prototype F_k est constitué par la liste des coefficients des fonctions d'intensité.

Voici la description de l'algorithme de segmentation d'une image $I(u, v)$:

étape 0 : Fixer le nombre de classes c et le type des fonctions $F_k(u, v)$ sur l'image (par exemple des polynômes). Initialiser ces fonctions $F_k(u, v)$ par une fonction constante avec une valeur aléatoire.

étape s : Générer une nouvelle partition de l'image en utilisant l'équation suivante du degré d'appartenance flou :

$$u_k(u, v) = \frac{1}{\sum_{j=1}^c \left(\frac{d(F_j(u, v), I(u, v))}{d(F_k(u, v), I(u, v))} \right)^{\frac{2}{m-1}}}$$

étape s' : Calculer les nouvelles fonctions sur l'image $F_k(u, v)$ qui minimisent l'erreur pondérée aux moindres carrés entre les points $(x, y, F_k(u, v))$ et $(x, y, I(u, v))$:

$$\sum_{(u,v) \in image} (u_k(u, v))^m d^2(F_k(u, v), I(u, v))$$

étape s'' : Si les fonctions sont stables, il faut s'arrêter, sinon retourner à l'étape s .

Lorsque l'algorithme a convergé, on dispose alors de c fonctions qui donnent la variation de l'intensité des images pour chaque classe. Par contre, la classification obtenue est floue, c'est-à-dire que l'on n'a pas une image de labels, mais c images de degrés d'appartenance $u_k(u, v)$. Pour néanmoins calculer l'image de label, on décide d'attribuer à chaque pixel la classe qui a le degré d'appartenance maximum.

4.2 Segmentation des cartes de profondeur

Dans la figure 15, le nombre de classes est volontairement petit, car sinon il est difficile de voir les différences entre l'image originale et l'image reconstituée à partir de la segmentation (un tel exemple est visible sur la figure 16). Cette image d'approximation est calculée en remplaçant le label par la valeur de la fonction quadratique associée au label, sur chaque pixel.

Pour la figure 15, les fonctions initiales sont constantes. La valeur de ces constantes est tirée aléatoirement dans l'intervalle de variation de la profondeur. Mais, une initialisation plus sophistiquée, du type [11, 55], permettrait d'accélérer la convergence, et aurait l'avantage de fournir des résultats plus proches de nos attentes sur ce type de données, lorsque le nombre de classes est plus grand.

Un exemple d'initialisation plus sophistiqué est visible sur la figure 16. Une première segmentation est réalisée avec trois classes qui ont des fonctions d'intensité constante. Les régions de grande taille obtenues sont alors au nombre de six (figure 16(c)). Cette image de label permet d'initialiser l'algorithme de segmentation avec des régions à profil quadratique, pour aboutir à une approximation très proche de l'image originale (figure 16(b) et (f)).

4.3 Segmentation des images d'intensité

La figure 17 montre la segmentation de l'image d'un visage par des morceaux de fonctions quadratiques. On remarque, sur la série d'images avec des orientations différentes, que cette méthode permet de localiser et de suivre les yeux et la bouche sur cette séquence.

Une autre application de cette méthode consiste à utiliser l'image de labels et les équations des classes pour reconstituer une approximation de l'image. La bonne proximité entre les images d'approximation et originales permet d'envisager l'ap-

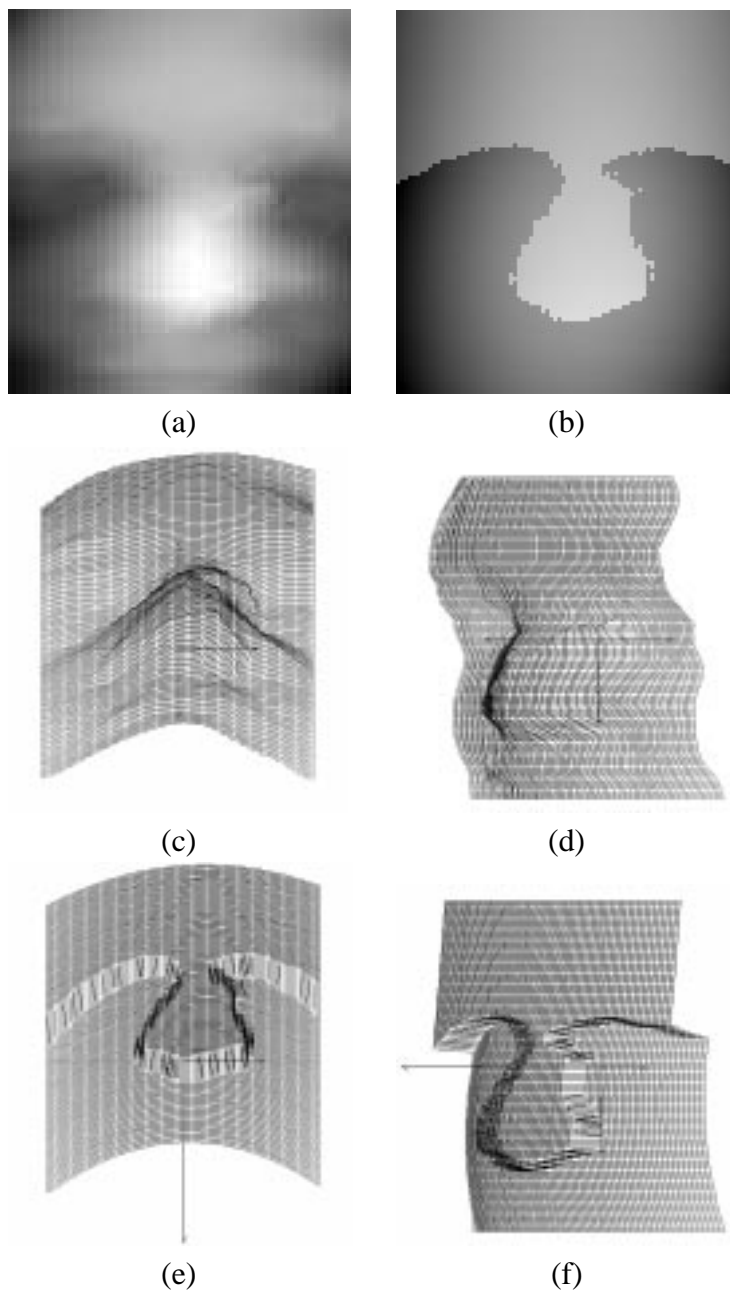


FIG. 15 – (a) est la carte de profondeur originale, dont on peut voir deux vues 3D en (c) et (d). (b) est l'approximation par deux fonctions quadratiques (observer les courbures inverses de la région supérieure : front et nez) obtenue par l'approche MMP. (e) et (f) ont les mêmes points de vue que (c) et (d), mais elles présentent l'approximation quadratique.

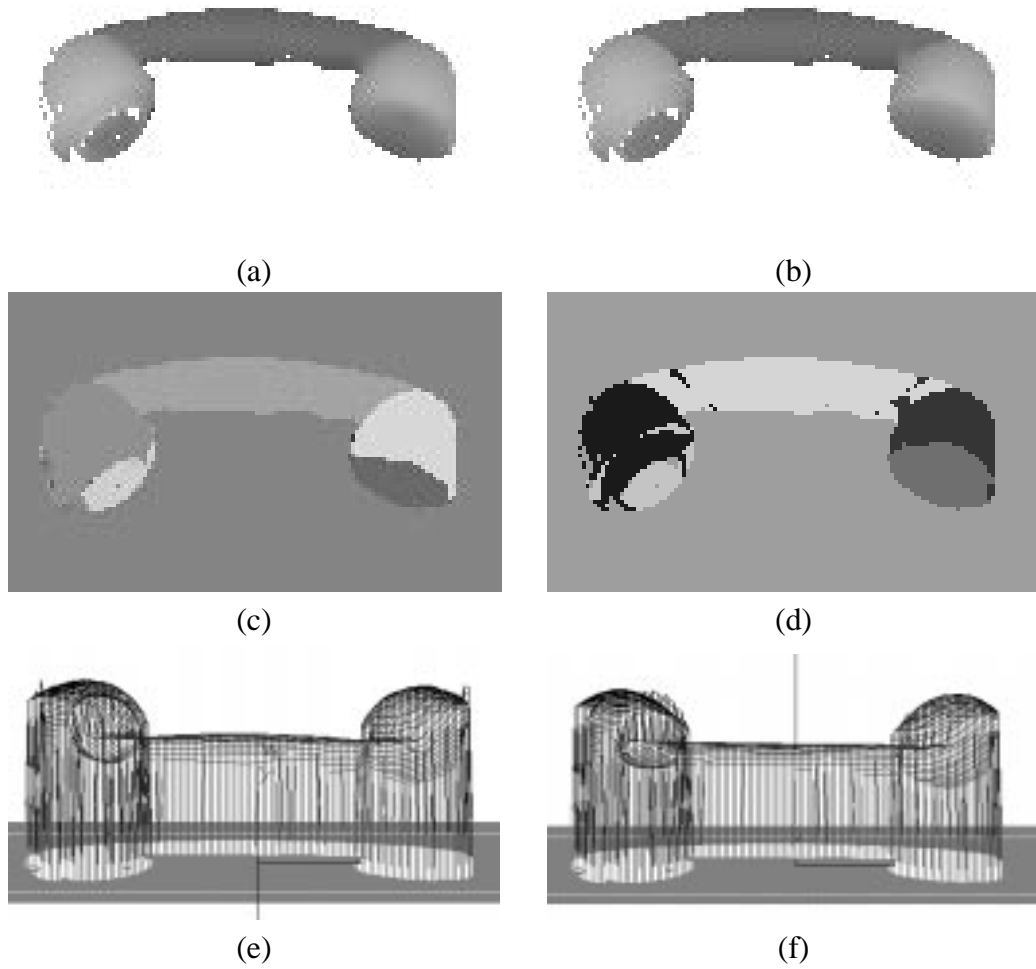


FIG. 16 – (a) est la carte de profondeur originale d'un combiné de téléphone, dont une vue 3D est en (e). (b) est l'approximation par cinq fonctions quadratiques obtenue par la méthode MMP dont une vue 3D est en (f). (d) est l'image de labels finale et (c) est l'image de labels initiale qui est obtenue par segmentation avec des régions de profondeur constante.



FIG. 17 – Les images originales d'un visage avec différentes orientations sont sur la première ligne. Sur la deuxième ligne, ces images sont segmentées en 5 classes qui ont une variation quadratique de l'intensité. La troisième ligne présente les images de labels associées. Sur la quatrième ligne, la segmentation du visage vu de face, avec 5, 4, 3 et 2 classes est visible. Les images de label associés sont sur la cinquième ligne.

plication de cette méthode à la compression des images en ne transmettant que les fonctions d'intensité et l'image de labels compressée par une technique standard.

4.4 Segmentation d'images couleur

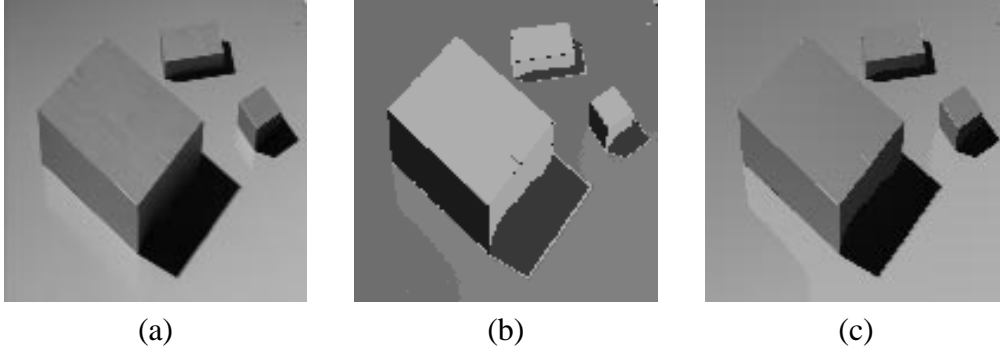


FIG. 18 – (a) *image originale en couleur.* (b) *image de labels de la segmentation avec 6 classes.* (c) *image couleur approximée linéairement.*

La section précédente présente des résultats de segmentation avec des images en niveau de gris. Mais la prise en compte de la couleur est une extension simple. En effet, sur la même surface, quelque soit le canal, l'intensité varie de la même façon (b et c). Ce qui change pour chaque canal est donc juste le terme constant a de la fonction prototype qui approxime la variation de l'intensité. Ainsi dans le cas linéaire, les canaux Rouge, Vert et Bleu sont segmentés à l'aide de la fonction prototype suivante :

$$I_R(u, v) = a_R + bu + cv$$

$$I_V(u, v) = a_V + bu + cv$$

$$I_B(u, v) = a_B + bu + cv$$

Les caractéristiques d'un prototype affine sont donc (a_R, a_V, a_B, b, c) , en couleur, au lieu de (a, b, c) sans couleur.

La couleur permet d'obtenir des segmentations de meilleure qualité en traitant plus d'informations que les niveaux de gris. Comme l'illustre la figure 18, notre méthode MMP permet de segmenter correctement en facettes cette scène polyédrique.

4.5 Discussion

Nous avons présenté l'application de notre approche à la segmentation. Cette méthode présente des propriétés intéressantes de convergence et s'adapte facilement à différents types d'images : images d'intensité en niveaux de gris ou en couleurs, cartes de profondeur.

La classification induit que les régions sont de tailles équivalentes. Ceci est heureux car cela évite l'apparition de régions très petites et non-pertinentes. Néanmoins, il n'y a aucune contrainte dans cette méthode qui oblige les régions à avoir un contour assez régulier. C'est un défaut qu'il serait utile de corriger.

5 Application à plusieurs modèles physiques déformables

Dans le cas où l'on désire modéliser un objet 2D ou 3D dont la forme est trop complexe pour bien être approximée par des primitives géométriques simples de base, on recourt à un modèle paramétré de référence plus général, libre de se déformer comme les 'snakes' ou les splines [13, 23, 4]. Puisque ces modèles géométriques ont un grand nombre de degrés de liberté, ils ont de bonnes propriétés pour recouvrir les données. Mais quand ces dernières manquent, elles ont des difficultés pour les interpoler [53, 22]. Faire une bonne interpolation des données manquantes nécessite une contrainte globale sur la forme du modèle. Cette contrainte est souvent induite par l'utilisation d'une famille de surfaces ou courbes plus restreinte comme les super-quadriques et les super-ellipsoïdes [26, 8], des sous-ensembles de courbes algébriques [50], des modèles hybrides [12, 3], ou des modèles physiques déformables [31, 42, 40].

Nous présentons maintenant une dernière application de notre approche pour des modèles que nous nommons articulés, et qui permettent de faire du recalage fin de contours 2D polyédriques ou non, tout en étant contraints de façon globale sur leur forme.

5.1 Modèle déformable constitué de segments articulés

Ces modèles articulés, ou structures polygonales sont une variation des modèles physiques déformables. En effet, les modèles articulés sont aussi constitués d'un ensemble de points pouvant se déplacer, reliés par des segments.

Avec les modèles physiques déformables, les sommets du modèle sont attirés par le point des données le plus proche. Dans notre cas, c'est l'inverse puisque chaque point des données exerce sur les primitives articulées une force d'attraction qui est proportionnelle à l'éloignement. Cette inversion, a priori anodine, change la nature des problèmes qu'il est possible de traiter avec ces deux types de modèles. Cela a déjà été remarqué dans le cadre des 'snakes' sans être utilisé [38]. Par construction les modèles physiques, ou les 'snakes', peuvent se recaler finement sur une petite partie des contours à partir d'une position initiale bien choisie. Avec les modèles articulés, l'inclusion est dans l'autre sens. Les primitives articulées doivent donc recouvrir l'ensemble des contours (section 5.3). Cette contrainte est modulable en introduisant une classe de rejet qui recueille les points parasites. Un des avantages de modéliser toutes les données est de permettre une initialisation du modèle physique plus distante de la solution finale comme le montre la section 5.2.

Un autre avantage des primitives articulées sur les modèles physiques déformables est qu'ils permettent de s'affranchir d'une discrétisation fine du modèle, en pouvant aller jusqu'à une approximation polygonale d'un contour courbe. Les modèles physiques sont en effet réservés aux courbes, alors que les primitives articulées permettent de modéliser tous les types de contours, qu'ils soient polyédriques ou courbes.

Dans un modèle physique, chaque sommet du système subit une force vers le plus proche voisin du contour, qui est proportionnelle à sa distance. Dans notre cas, chaque point des contours de l'image induit une force sur la primitive déformable. Nous supposons donc qu'elle agit sur le plus proche voisin qui appartient à la primitive. Quand ce point est entre deux sommets de la primitive, la force est répartie sur les deux extrémités du segment en la pondérant par la distance au sommet. Par ce biais, nous nous ramenons à une action des données sur les sommets, comme pour les modèles physiques déformables. Cette modification était nécessaire pour pouvoir incorporer les modèles physiques dans le cadre général de l'algorithme MMP décrit dans la section 2.

De manière générale, le grand nombre de degrés de liberté d'un modèle de contour actif peut laisser la possibilité au contour de converger et de se déformer vers presque n'importe quoi sous l'influence des forces de proximité, en l'absence de contraintes supplémentaires. La contrainte sur un modèle déformable physique est induite par l'hypothèse que ses sommets sont reliés par des ressorts et que le système doit se comporter selon les lois de la physique, d'où son nom. Pour stabiliser les oscillations du système une force de frottement fluide est ajoutée. Une autre contrainte doit être utilisée pour faire de l'interpolation : la régularité de la forme.

Nous nous sommes inspirés de la démarche modèle déformable physique pour déplacer nos primitives articulées. Le but de ces contraintes est d'empêcher que la primitive initiale se déforme trop brutalement et converge alors vers des positions peu intéressantes. Pour autant, la primitive doit être très facilement déformable pour bien s'adapter aux contours. Or, ce type de comportement est caractéristique des liquides visqueux, qui sont infiniment déformables au repos, mais présentent une résistance forte aux déformations plus la vitesse est grande. Un liquide visqueux subit une force de frottement interne proportionnelle à la différence de vitesses de chaque point par rapport à ses voisins. Cette force est caractérisée par un coefficient de viscosité $v_{elastic}$. De plus, globalement le liquide est soumis à une force d'amortissement fluide de facteur a_{fluide} . Nous avons donc contraint notre primitive à se déformer dynamiquement comme un liquide visqueux par l'équation suivante :

$$\mu_{sommets} X''_i + a_{fluide} X'_i - v_{elastic}(X'_i - X'_{i+1}) - v_{elastic}(X'_i - X'_{i-1}) = \Phi_i \quad (21)$$

où Φ_i est la force qui s'exerce sur le sommet X_i . $\mu_{sommets}$ est la masse des sommets. Cette équation est résolue itérativement par un schéma d'intégration d'Euler.

Nous avons présenté, pour l'instant, qu'une variante des contours déformables physiques. Voici à présent comment on arrive à intégrer cette variante dans le cadre de la modélisation avec plusieurs prototypes, pour disposer d'une méthode de contours actifs multiples.

Il est clair que le calcul de la force est une autre façon de faire de l'appariement implicitement, comme avec le plus proche voisin. La condition pour que le schéma de l'algorithme MMP fonctionne est que les étapes d'appariement et d'estimation diminuent la distance entre les prototypes et les données. Par analogie, on peut considérer qu'une itération du schéma d'intégration d'Euler de l'équation dy-

namique (21) du contour actif diminue l'écart entre le modèle et les données. Cette itération va donc remplacer l'étape d'estimation géométrique.

Ainsi, une classe correspond à une primitive articulée. Cette dernière a une structure physique bien déterminée par la façon dont sont choisies les connections entre les sommets. Toutefois, chaque sommet de la primitive peut se déplacer. Ainsi, les coordonnées des sommets forment l'ensemble des degrés de liberté de la primitive. Les prototypes F_k sont donc constitués de la liste des coordonnées des sommets de la primitive articulée k (cf. figure 19).

L'algorithme de recalage multiple des contours articulés se résume donc à :

étape 0 : Fixer le nombre de classes c et la structure des primitives articulées. Initialiser les sommets des primitives F_k interactivement. (*Donner la distance δ .*).

étape a : Calculer le plus proche voisin X'_{ik} appartenant à la primitive F_k de chaque point X_i . Ajouter à chaque sommets du segment de la primitive F_k qui contient X'_{ik} , la force $\Phi = X_i - X'_{ik}$ pondérée respectivement par la distance de X'_{ik} aux sommets du segment. (*seulement pour k entre 1 et $c - 1$*).

étape a' : Générer une nouvelle partition en utilisant l'équation suivante du degré d'appartenance flou :

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d(X_i, X'_{ik})}{d(X_i, X'_{ij})} \right)^{\frac{2}{m-1}}}$$

(où $d(X_i, X'_{ic}) = \delta$).

étape a'' : Calculer les nouveaux sommets des primitives F_k en résolvant une itération de l'équation (21) sur l'ensemble des sommets. (*seulement pour k entre 1 et $c - 1$*).

étape a''' : Si les sommets des contours actifs sont stables, il faut s'arrêter, sinon retourner à l'étape a.

5.2 Recalage fin sur des contours polyédriques

Une des qualités des contours déformables, donc des primitives articulées, est qu'il est possible de choisir la structure du modèle en fonction de l'application.

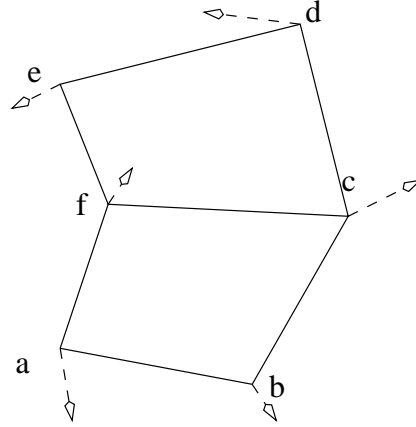


FIG. 19 – Structure en 8 digital décrite par ses six sommets (a, b, c, d, e, f) et ses sept arrêtes $(a - b)$, $(b - c)$, $(c - d)$, $(d - e)$, $(e - f)$, $(f - a)$ et $(c - f)$. Chaque sommet peut se déplacer en 2D, d'où 12 degrés de linerté.

Ainsi, sur les images aériennes des figures 20 et 21, il est clair qu'un modèle convenable d'une toiture de maison à deux pentes est une structure en 8 digital (figure 19). La structure géométrique de cette forme de contour est simple à décrire par des listes d'association entre segments et sommets.

Pour obtenir les résultats des figures 20 et 21, nous avons initialisé le paramètre a_{fluide} d'amortissement à 0.1, la viscosité $v_{elastic}$ à 2.5, la masse μ_{sommet} à 1. Le pas de temps Δt dans la méthode d'Euler est 0.1. Ces paramètres sont nombreux, mais relativement fixes. Seul le pas Δt doit être diminué en cas de non convergence. Le paramètre δ qui détermine l'influence d'un point est plus délicate à régler, car il définit implicitement la taille de la classe des points rejetés (avec le facteur flou m toujours choisi égal à 1.5). La valeur de $\delta = 2$ pixels, obtenue par des essais, nous a donné satisfaction dans les deux cas.

Néanmoins, une initialisation assez proche du contour du toit est nécessaire, pour ne pas converger vers les contours parasites voisins des toits qui sont assez denses sur ces images.

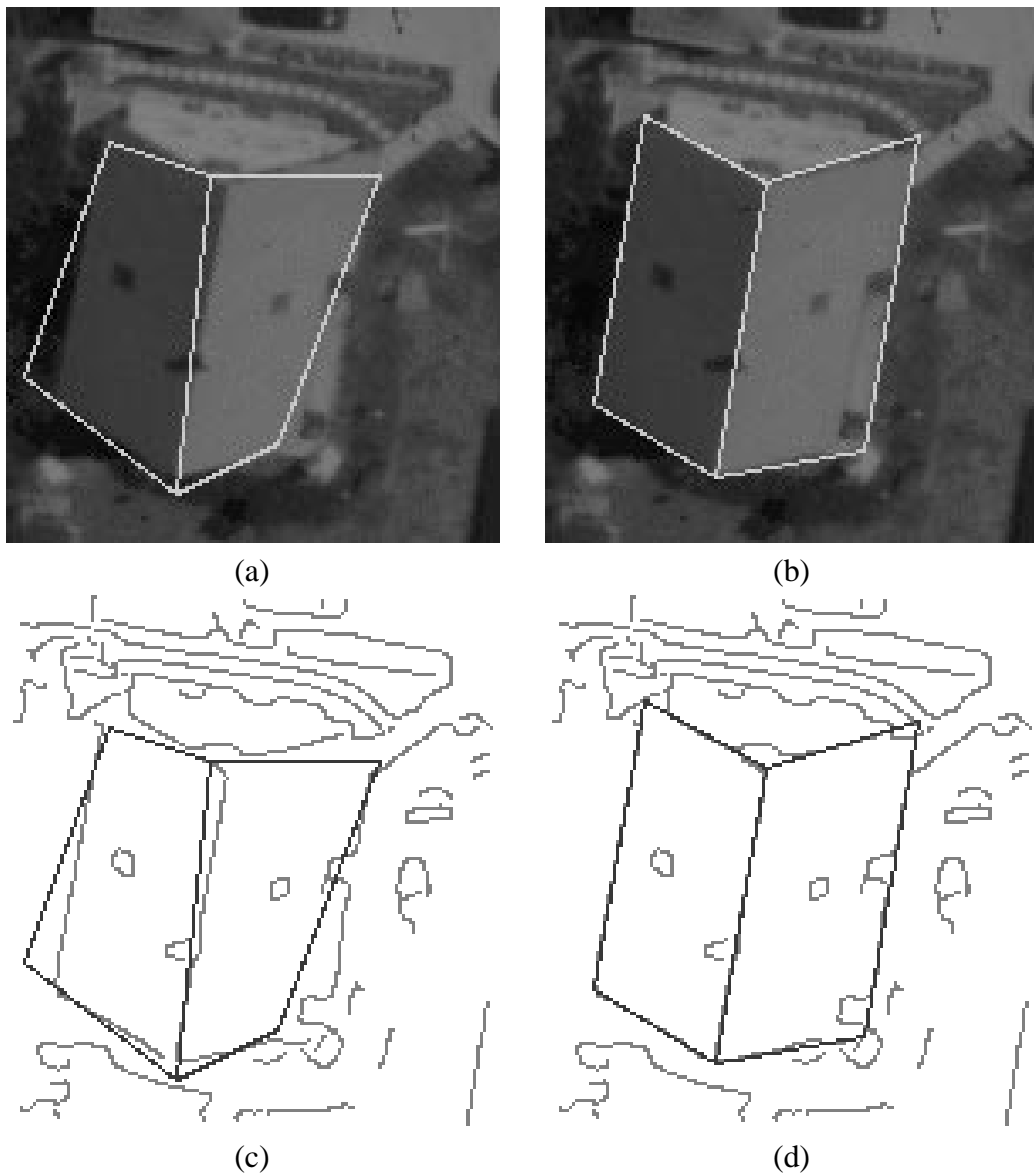


FIG. 20 – *Superposition, sur une image aérienne d’une résidence de la primitive articulée initiale (a) et finale (b). (c) et (d) montrent la même superposition sur les contours qui ont servi au recalage.*

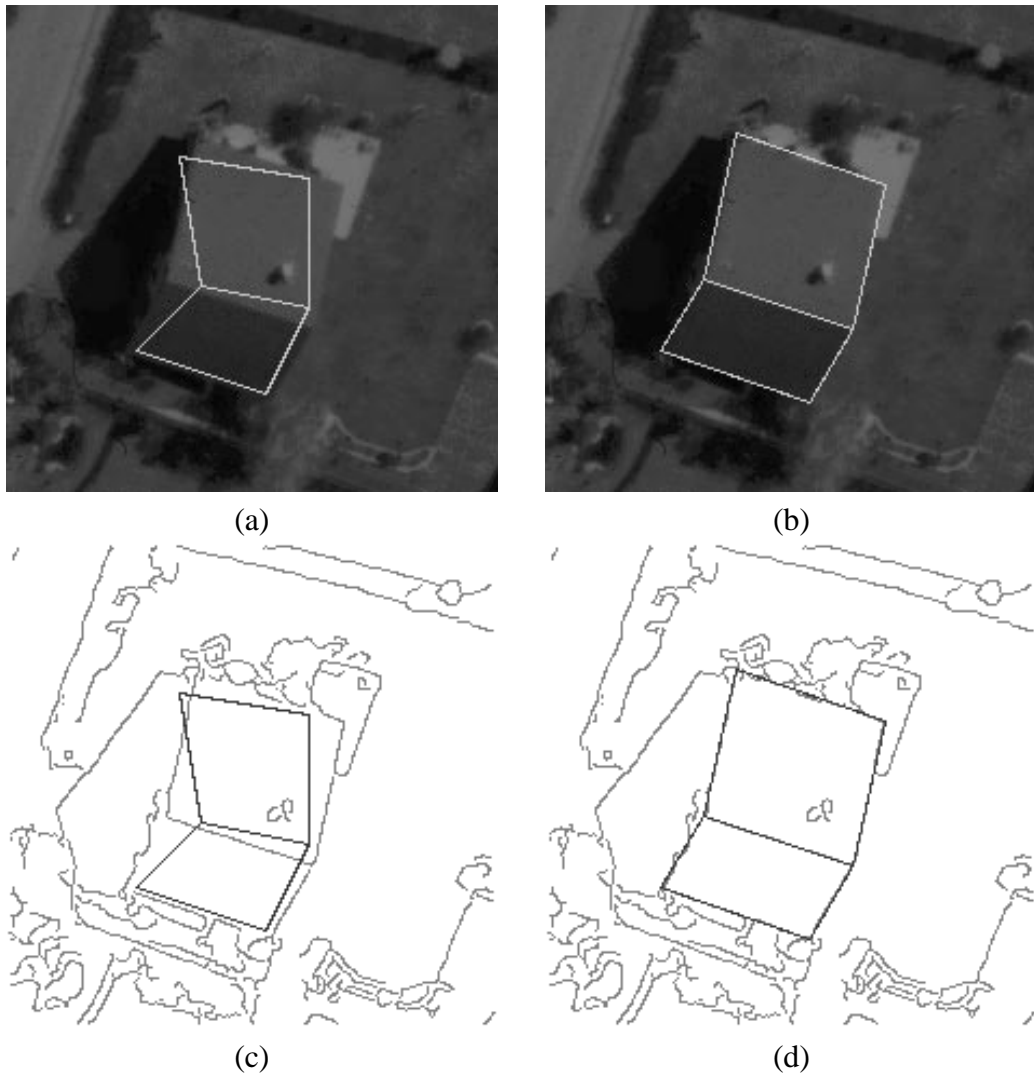


FIG. 21 – *Superposition, sur une image aérienne d’une résidence de la primitive articulée initiale (a) et finale (b). (c) et (d) montrent la même superposition sur les contours qui ont servi au recalage.*

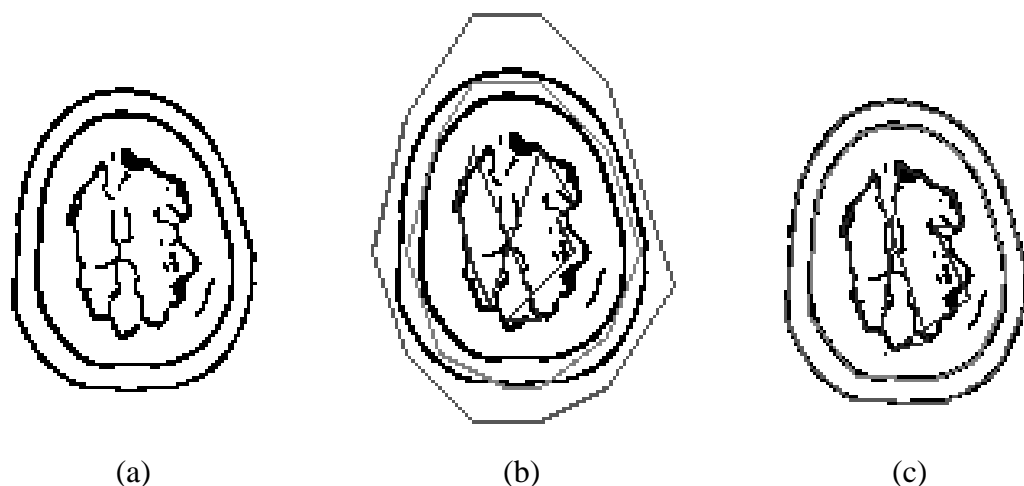


FIG. 22 – (a) est la coupe originale du crâne qui présente trois contours : l'extérieur, l'intérieur du crâne, et le contour du cerveau. (b) montre l'initialisation par trois chaînes de 10 segments qui est relativement approximative. (c) est le résultat après recalage fin réalisé en parallèle.

5.3 Recouvrement par plusieurs contours

La figure 22 illustre la possibilité de faire du recalage fin avec plusieurs contours actifs en même temps. Avec les approches habituelles, qui font l'appariement des sommets du contour déformable avec le plus proche voisin ou utilisent un champ de force sur l'image dérivée du gradient, la définition du modèle initial pour obtenir le contour central est assez délicate dans cet exemple. En effet, l'initialisation doit être suffisamment proche de la paroi intérieure du crâne pour que le contour actif ne soit pas attiré par les autres contours. Par contre, puisque notre algorithme détermine à chaque étape l'appartenance d'un point du contour à une primitive articulée ou à une autre, l'initialisation peut être beaucoup moins précise et osciller entre les parties intérieure et extérieure du crâne comme dans la figure 22(b). Il n'est toutefois pas possible de s'écarter beaucoup plus sans risquer de dégrader les résultats.

La condition de cette plus grande robustesse est la possibilité de recouvrir l'ensemble des contours par des primitives articulées même si elles sont dégénérées.

5.4 Discussion

Nous avons donc présenté dans cette section, l'application de notre approche de modélisation multi-prototypes aux contours actifs dans le but de pouvoir modéliser des objets de forme plus compliquée. Cela nous a conduit à introduire un nouveau type de modèle : la primitive articulée qui est une structure formée de segments dont les sommets se comportant physiquement comme un fluide visqueux. Cette primitive présente l'avantage de permettre de modéliser aussi bien les contours courbes que ceux polyédriques. Enfin, l'introduction de l'approche MMP permet d'améliorer la robustesse des modèles physiques à l'initialisation.

6 Conclusion

Nous avons présenté une généralisation (GICP) de la méthode ICP à plusieurs importantes transformations géométriques comme les similitudes et les homographies. Mais, comme cet algorithme ne peut manipuler qu'un unique prototype, nous avons proposé un nouvel algorithme pour faire la modélisation avec plusieurs prototypes. Par là, nous avons abouti à la conception d'une méthode générale de Modélisation Multi-Prototypes (MMP).

La MMP est une méthode pour résoudre le problème du recouvrement ou de modélisation d'un ensemble de données par des prototypes géométriques déformables. En effet, il s'inspire d'une méthode de classification avec ou sans points parasites, et de la version pondérée de l'algorithme GICP pour produire des résultats robustes aux points aberrants et aux positions bruitées des points. La description de l'algorithme MMP est très simple. Néanmoins, il est puissant et général, car il combine itérativement une étape d'appariement et une étape d'estimation. De plus, il est contrôlé par seulement deux paramètres importants : le nombre de classes c et la distance δ des points probablement parasites. Le nombre de classes est un paramètre de contrôle efficace. Mais, pour certaines applications, il est intéressant d'étudier, à l'avenir, comment on peut déterminer ce paramètre automatiquement.

Enfin, nous voulons insister sur le fait que notre approche qui consiste à mélanger classification et recalage avec appariement vers le plus proche voisin est une voie de recherche prometteuse. Nous avons montré que ce type d'approche peut être appliqué avec succès sur des fonctions pour faire de la segmentation d'images, et sur des structures articulées, variantes des modèles physiques déformables pour

faire du recalage fin de contours 2D déformables. L'application aux splines semble aussi possible. Une autre application, à des surfaces maillées déformables, est aussi envisageable dans un esprit proche des travaux de [23].

Si la généralisation à des projections en dimension p de prototypes en dimension q pose des problèmes de stabilité, c'est un sujet d'études intéressant par ces nombreuses applications, en particulier pour le suivi en parallèle de plusieurs objets dans une séquence d'images.

Références

- [1] Ackah-Miezan (A.) et Gagalowicz (A.). – Discrete models for energy minimizing segmentation. In : *Fourth International Conference on Computer Vision (Berlin, Germany, May 11–14, 1993)*. pp. 200–207. – Washington, DC, 1993. <http://www-rocq.inria.fr/syntim/textes/iccv93-eng.html>.
- [2] Bajcsy (R.), Solina (F.) et Gupta (A.). – *Segmentation versus Object Representation—Are They Separable?*, pp. 207–223. – Springer-Verlag, 1990.
- [3] Bardinet (E.), Cohen (L.D.) et Ayache (N.). – *A Parametric Deformable Model to Fit Unstructured 3D Data*. – Rapport technique n° 2617, INRIA, July 1995.
- [4] Bascle (B.) et Deriche (R.). – Stereo matching, reconstruction and refinement of 3-D curves using deformable contours. In : *Fourth International Conference on Computer Vision (Berlin, Germany, May 11–14, 1993)*. – Washington, DC, 1993.
- [5] Besl (P.J.) et McKay (N.D.). – A method for registration of 3D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, n° 2, February 1992, pp. 239–256.
- [6] Bezdek (J.C.). – *Pattern Recognition with Fuzzy Objective Function Algorithms*. – Plenum Press, 1981.
- [7] Bolle (R.M.) et Cooper (D.B.). – Bayesian recognition of local 3-D shape by approximating image intensity functions with quadric polynomials. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, n° 4, July 1984, pp. 418–429.

- [8] Bolle (R.M.) et Vemuri (B.C.). – On three-dimensional surface reconstruction methods. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, n° 1, January 1991, pp. 1–13.
- [9] Boufama (B.), Mohr (R.) et Veillon (F.). – Euclidean constraints for uncalibrated reconstruction. In : *Fourth International Conference on Computer Vision (Berlin, Germany, May 11–14, 1993)*. pp. 466–470. – Washington, DC, 1993.
- [10] Boujemaa (N.). – *Modélisation floue de l'incertitude pour la segmentation d'images*. – Thèse de PhD, Université Paris-V, 1993.
- [11] Bricault (I.) et Monga (O.). – *From volume medical images to quadratic surface patches*. – Rapport technique n° 2380, INRIA, September 1994.
- [12] Cohen (I.) et Cohen (L.D.). – *A Hybrid hyperquadric model for 2-D and 3-D data fitting*. – Rapport technique n° 2188, INRIA, January 1994.
- [13] Cohen (I.), Cohen (L.D.) et Ayache (N.). – *Introducing deformable surfaces to segment 3D images and infer differential structures*. – Rapport technique n° 1403, INRIA, March 1991.
- [14] Dave (R.N.). – Characterization and detection of noise in clustering. *Pattern Recognition Letters*, vol. 12, 1991, pp. 657–664.
- [15] Dave (R.N.) et Bhaswan (K.). – Adaptive fuzzy c-shells clustering and detection of ellipses. *IEEE Trans. Neural Networks*, vol. 3, n° 5, 1992, pp. 643–662.
- [16] Deriche (R.) et Faugeras (O.). – *Les EDP en traitement des images et vision par ordinateur*. – Rapport technique n° 2697, INRIA, November 1995.
- [17] Eklundh (J.O.), Yamamoto (H.) et Rosenfeld (A.). – A relaxation method for multispectral pixel classification. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, n° 1, January 1980, pp. 72–75.
- [18] Faugeras (O.) et Hebert (M.). – The representation, recognition and positioning of 3D shapes from range data. In : *Proceedings of the 8th International Conference On Artificial Intelligence*, pp. 996–1002. – Karlsruhe, BRD, August 1983.

- [19] Faugeras (O.), Hébert (M.) et Pauchon (E.). – Segmentation of range data into planar and quadratic patches. In : *Proceedings, CVPR '83 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Arlington, VA, June, 1983)*. pp. 8–13. – IEEE.
- [20] Feldmar (J.) et Ayache (N.). – Locally affine registration of free- form surfaces. In : *CVPR'94 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, June 21-23, 1994)*. pp. 496–501. – Seattle, WA, June 1994.
- [21] Ferrie (F.P.), Lagarde (J.) et Whaite (P.). – Darboux frames, snake, and superquadrics: Geometry from the bottom-up. In : *Proceedings of Workshop on Interpretation of 3D Scenes*, pp. 170–176. – Austin, TX, 1989.
- [22] Fua (P.) et Brechbuhler (C.). – Imposing hard constraints on soft snakes. In : *Proceedings, ECCV '96 (Fourth European Conference on Computer Vision, Cambridge, England, April 1996)*, pp. 495–506.
- [23] Fua (P.) et Leclerc (Y.G.). – Registration without correspondences. In : *CVPR'94 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, June 21-23, 1994)*. pp. 121–128. – Los Alamitos, CA, June 1994.
- [24] Fua (P.) et Sander (P.T.). – Segmenting unstructured 3D points into surfaces. In : *Proceedings, ECCV '92 (Second European Conference on Computer Vision, Santa Margherita Ligure, Italy, May 1992)*, pp. 676–680.
- [25] Gagalowicz (A.) et Monga (O.). – A new approach to image segmentation. In : *Proceedings, Eighth International Conference on Pattern Recognition (Paris, France, October 27–31, 1986)*. pp. 265–267. – Washington, DC, 1986.
- [26] Gross (A.D.) et Boulton (T.E.). – Error of fit measures for recovering parametric solids. In : *Second International Conference on Computer Vision (Tarpon Springs, FL, December 5–8, 1988)*. pp. 690–694. – Washington, DC, 1988.
- [27] Gupta (A.) et Bajcsy (R.). – Volumetric segmentation of range images of 3D objects using superquadric models. *CVGIP - Image Understanding*, vol. 58, n° 3, 1993, pp. 302–326.

-
- [28] Haralick (R.M.) et Shapiro (L.G.). – Image segmentation techniques. *Computer Vision, Graphics and Image Processing*, vol. 29, n° 1, January 1985, pp. 100–132.
- [29] Jolion (J.M.), Meer (P.) et Bataouche (S.). – Robust clustering with applications in computer vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, n° 8, August 1991, pp. 791–802.
- [30] Kanatani (K.). – Analysis of 3D roation fitting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, n° 5, May 1994, pp. 543–549.
- [31] Kass (M.), Witkin (A.) et Terzopoulos (D.). – Snakes: Active contour models. *International Journal of Computer Vision*, vol. 1, 1987, pp. 321–331.
- [32] Koepfler (G.), Lopez (C.) et Morel (J.M.). – A multiscale algorithm for image segmentation by variational method. *SIAM J. Numerical Analysis*, vol. 31, 1994, pp. 282–299.
- [33] Koivunen (V.), Vézien (J-M.) et Bajcsy (R.). – Procedural CAD-models from range data. In : *Proceedings of Towards World Class Manufacturing '93*.
- [34] Krishnapuram (R.) et Freg (C.P.). – Fitting an unknown number of lines and planes to image data through compatible cluster merging. *Pattern Recognition*, vol. 25, n° 4, 1992, pp. 385–400.
- [35] Krishnapuram (R.), Nasraoui (O.) et Frigui (H.). – The fuzzy c-spherical shells algorithm: A new approach. *IEEE Trans. Neural Networks*, vol. 3, n° 5, 1992, pp. 663–670.
- [36] Leclerc (Y.G.). – Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, vol. 3, 1989, pp. 73–102.
- [37] Lengagne (Richard), Tarel (Jean-Philippe) et Monga (Olivier). – From 2d images to 3d face geometry. In : *Proceedings of IEEE Second International Conference on Automatic Face and Gesture Recognition*. – Killington, USA, 1996. <http://www-rocq.inria.fr/syntim/textes/fg96-eng.html>.

- [38] Liao (C.W.) et Medioni (G.). – Simultaneous segmentation and approximation of complex patterns. *In: CVPR'94 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, June 21-23, 1994)*. pp. 617–623. – Los Alamitos, CA, June 1994.
- [39] Mukherjee (J.), Das (P.P.) et Chatterji (B.N.). – Segmentation of range images. *Pattern Recognition*, vol. 25, n° 10, 1992, pp. 1141–1156.
- [40] Nastar (C.). – *Modèles physiques déformables et modes vibratoires pour l'analyse du mouvement non-rigide dans les images multidimensionnelles*. – Thèse de PhD, École Nationale des Ponts et Chaussées, 1994.
- [41] Pentland (A.). – Recognition by parts. *In: First International Conference on Computer Vision, (London, England, June 8–11, 1987)*. pp. 612–620. – Washington, DC, 1987.
- [42] Pentland (A.) et Sclaroff (S.). – Closed-form solutions for physically based shape modelling and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, n° 7, July 1991, pp. 715–729.
- [43] Randriamasy (S.). – *Segmentation descendante coopérative en régions de paires d'images stéréoscopiques*. – Thèse de PhD, Université Paris-IX Dauphine, 1992.
- [44] Stockman (George). – *Object Recognition*, pp. 225–253. – Springer-Verlag, 1990.
- [45] Tarel (Jean-Philippe). – Multi-objets interpretation. *In: Proceedings, 13th International Conference on Pattern Recognition (ICPR13, Vienna, Austria, August 25-30, 1996)*. pp. 612–616. – Computer Society Press. <http://www-rocq.inria.fr/syntim/textes/icpr96-tarel-eng.html>.
- [46] Tarel (Jean-Philippe). – *Reconstruction globale et robuste de facettes 3D*. – Rapport de recherche n° 2813, INRIA, February 1996. <http://www-rocq.inria.fr/syntim/textes/RR-2813-fra.html>.
- [47] Tarel (Jean-Philippe) et Boujemaa (Nozha). – *Une approche floue du recalage 3D : généricité et robustesse*. – Rapport de recherche n° 2716, INRIA, 1995.

-
- [48] Tarel (Jean-Philippe) et Boujemaa (Nozha). – Une approche floue du recalage 3D : généricité et robustesse. In : *10ème congrès AFCET, Reconnaissance des Formes et Intelligence Artificielle*. – Rennes, France, 1996. <http://www-rocq.inria.fr/syntim/textes/rfia96-fra.html>.
- [49] Taubin (G.). – Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, n° 11, November 1991, pp. 1115–1138.
- [50] Taubin (G.). – An improved algorithm for algebraic curve and surface fitting. In : *Fourth International Conference on Computer Vision (Berlin, Germany, May 11–14, 1993)*. pp. 658–665. – Washington, DC, 1993.
- [51] Vézien (J.M.). – *Techniques de reconstruction globale par analyse de paires d'images stéréoscopiques*. – Thèse de PhD, Université Paris-VII, 1995.
- [52] Vinet (L.). – *Segmentation et mise en correspondance de régions de paires d'images stéréoscopiques*. – Thèse de PhD, Université Paris-IX Dauphine, 1991.
- [53] Weiss (I.). – Shape reconstruction on a varying mesh. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, n° 4, April 1990, pp. 345–362.
- [54] Xie (X.L.) et Beni (G.). – A validity measure for fuzzy clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 13, n° 8, August 1991, pp. 841–847.
- [55] Yokoya (N.) et Levine (M.D.). – Range image segmentation based on differential geometry: A hybrid approach. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 11, n° 6, June 1989, pp. 643–649.
- [56] Zhang (J.) et Modestino (J.W.). – A model-fitting approach to cluster validation with application to stochastic model-based image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, n° 10, October 1990, pp. 1009–1017.

- [57] Zhang (Z.). – Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, vol. 13, n° 2, 1994, pp. 119–152. – also INRIA Tech. Report #1658.
- [58] Zhang (Z.Y.), Luong (Q.T.) et Faugeras (O.). – Motion of an uncalibrated stereo rig: Self-calibration and metric reconstruction. *IEEE Trans. Robotics and Automation*, vol. 12, n° 1, February 1996, pp. 103–113.



Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399