Contributions aux méthodes variationnelles pour le calcul du flot optique: prise en compte des informations spatiales et temporelles

Dominique Béréziat

UNIVERSITÉ PIERRE ET MARIE CURIE

HABILITATION À DIRIGER DES RECHERCHES

Contributions aux méthodes variationnelles pour le calcul du flot optique : prise en compte des informations spatiales et temporelles

par Dominique BÉRÉZIAT

soutenue le 24 novembre 2010 devant le jury constitué de :

François-Xavier Le Dimet	Professeur émérite à l'Université Joseph Fourier	Rapporteur
Jocelyn Chanussot	Professeur à l'Institut de technologie de Grenoble	Rapporteur
Gérard Médioni	Professeur à University of Southern California	Rapporteur
Laure Blanc-Féraud	Directrice de recherche CNRS à Sophia-Antipolis	Examinatrice
Jean-Luc Zarader	Professeur à l'Université Pierre et Marie Curie	Examinateur
René Alt	Professeur émérite à l'Université Pierre et Marie Curie	Examinateur

Remerciements

On le dit souvent : cette page est sans doute la plus difficile à écrire et je peux confirmer que c'est bien le cas tant le nombre de personnes à qui je dois ce manuscrit, de manière directe ou indirecte, est grand. Ceci illustre aussi le fait que la recherche scientifique est avant tout un travail collectif qui dépasse largement le cadre du laboratoire, de l'établissement et même de la nation. D'avance, je présente mes excuses auprès de ceux et celles dont le nom n'apparaît dans cette page.

En tout premier lieu, je voudrais remercier François-Xavier Le Dimet, Gérard Médioni et Jocelyn Chanussot qui m'ont fait un grand honneur en ayant accepté de rapporter ces travaux en un temps relativement court et cela malgré leurs fortes contraintes. Merci également à Jean-Luc Zarader et Laure Blanc-Féraud qui ont accepté d'être examinateurs dans ce jury.

En second lieu, je veux remercier mes collègues ici à l'Université Pierre et Marie Curie : René Alt, qui m'a accueilli dans son équipe en 2001 et qui m'a donné sa confiance pour initier des travaux autour du traitement d'images et bénéficier également de la synergie de l'UPMC autour de l'imagerie biologique ; Séverine Dubuisson, avec qui nous avons, ensemble, essayé de pérenniser ces activités de recherche (et aussi d'enseignement), malgré parfois les oppositions ou les inerties. Sachant que j'oublie de nombreux collègues du laboratoire d'informatique de Paris 6, qu'ils soient (enseignants-)chercheurs, ingénieurs ou administratifs, je me bornerai à remercier les membres de mon équipe : Pierre Fortin, Stef Graillat, Valérie Menissier-Morain, Fabienne Jézéquel, Jean-Luc Lamotte, Laurent-Stéphane Didier, Jean Vignes et Jean-Claude Bajard avec lesquels j'ai beaucoup appris en arithmétique des ordinateurs et en calcul scientifique haute performance. Je remercie également et surtout les doctorants dont j'ai participé à l'encadrement : Wafa Rekik, Abir El Abed, Raimana Teina. Ils ont, eux aussi, contribué de manière importante à ce que ce manuscrit voit le jour.

Je ne remercierai jamais assez Isabelle Herlin pour m'avoir accueilli en délégation dans son équipe : je lui dois mes recherches sur l'assimilation de données pour le traitement des images, un terrain encore vierge avec des défis excitants à relever. Merci également aux autres membres de son équipe : Nicolas Mercier, Jean-Paul Berroir, Étienne Huot, Vivien Mallet, Nathalie Gaudechoux et tous les doctorants et post-doctorants.

Enfin, ces ultimes remerciements iront à ma famille : ma femme Christelle, mes magnifiques enfants Dorian et Léandre, j'espère avoir pu rester supportable et disponible malgré les contraintes et la charge de travail d'un enseignant-chercheur universitaire ; mes parents, Françoise et Gilbert, qui m'ont toujours soutenu et encouragé dans mes choix tout au long de ma vie ; mon frère Olivier pour son aide bricolage (passée et future) et ma sœur Véronique pour ses encouragements (elle aussi est enseignant-chercheur) ; mes beaux parents Jeannine et Serge, qui ont invariablement répondu présents pour nous et nos enfants. Et pour finir je n'oublierai pas tous mes amis, l'amitié est une chose trop importante pour être négligée.

Table des matières

Introduction générale

Dans ces travaux, nous nous intéressons aux problèmes mal posés en traitement d'images en général et au problème de l'estimation du mouvement apparent, le *flot optique*, dans une séquence d'images en particulier. Le problème de la reconstruction 3D est également abordé mais il est formulé en terme de compensation de mouvement. L'estimation du mouvement est, comme la plupart des problèmes du traitement d'images, mal posée [**?**]. En effet, le déplacement des points dans l'image est modélisé par une équation de conservation de la luminosité. Or, une telle équation est scalaire. La vitesse est un vecteur à deux composantes (dans le plan image), il y a donc une infinité de solutions. Une seconde difficulté est que l'équation de conservation de la luminosité est non linéaire par rapport à la vitesse, quantité que l'on souhaite calculer. On notera qu'il existe d'autres modèles pour représenter le déplacement des points dans l'image, modèles qui tirent parti de la modalité d'acquisition et du contexte d'étude. Par exemple, nous avions proposé un modèle spécifique aux images infrarouges satellitaires basé sur un principe de conservation de la masse [**?**]. Les présents travaux constituent donc un prolongement naturel de ces premières recherches. Quel que soit le modèle considéré, le problème de l'estimation du mouvement reste mal posé et non linéaire (c'est aussi le cas de nombreux problèmes du traitement d'images) et la question qui se pose est : quelles sont les informations supplémentaires nécessaires pour lever l'indétermination ?
Les dispositifs d'acquisition moderne ont maintenant la capacité de délivrer des données avec un rythme d'acquisition rapide. On dispose alors de séquences d'images dans lesquelles l'information temporelle, c'est-à-dire l'évolution en temps des structures spatiales observées dans une acquisition, peut être décrite à l'aide d'heuristiques. Il apparaît alors évident qu'une telle information sera utile pour la résolution d'un problème de traitement d'images donné quel qu'il soit. Toutefois ce type d'information reste sous utilisé pour deux raisons :

1. comment modéliser cette évolution temporelle ?
2. comment exploiter ce modèle dans les algorithmes existant ?

Dans ce manuscrit, nous utiliserons le formalisme du calcul des variations pour résoudre le problème posé et nous le déclinerons sous plusieurs formes et pour différentes problématiques. De façon très générale, un modèle liant les observations (les images) avec le vecteur d'état (les quantités que l'on souhaite calculer) est établi sous la forme d'une ou plusieurs équations, nous l'appellerons "modèle image". On construit alors une fonctionnelle de coût, dépendant du

vecteur d'état qui mesure la déviation du modèle à la réalité sur l'ensemble de l'image (ou de la séquence d'images). Le minimum de la fonctionnelle, qui fournit une solution ayant l'erreur la plus petite, est déterminé en calculant la différentielle de la fonctionnelle puis en résolvant l'équation d'Euler-Lagrange associée. On peut trouver un calcul complet dans [**?**], annexe A.1 de ce manuscrit, pour le calcul du flot optique. Si le problème est mal posé, il faut ajouter des contraintes supplémentaires, la technique éprouvée étant de borner les variations du vecteur d'état : on parle de régularisation [**?**]. On peut aussi ajouter d'autres contraintes qui tirent profit de l'information *a priori* lorsqu'elle est identifiable. Le problème n'étant plus sous-contraint on peut alors déterminer une solution unique.

C'est donc dans ce contexte que notre travail s'inscrit : résoudre des problèmes mal posés et non linéaires en traitement d'images. Résoudre un problème mal posé, hormis la régularisation, nécessite de prendre en compte l'information *a priori* pour formaliser une nouvelle contrainte liant les observations et le vecteur d'état. L'information *a priori* peut alors être de deux types :

1. spatiale : on fait des hypothèses structurelles sur la scène observée dans les images en incorporant des propriétés spatiales. Ce type d'approche est très utilisé en imagerie médicale et biologique puisqu'on peut tirer profit du fait qu'on connaît bien le contexte dans lequel se déroule l'acquisition. Nous l'illustrerons dans un cas particulier en imagerie cellulaire où l'on observe des structures sphériques. Nous exploiterons alors cette sphéricité pour réaliser une reconstruction 3D et également pour améliorer des algorithmes 2D tels que la segmentation ou le suivi d'objets évoluant à la surface de ces structures sphériques.
2. temporelle : si la dynamique de l'image est connue, nous la représentons sous la forme d'une *équation d'évolution*. Nous montrerons que la connaissance d'une telle équation et d'une condition initiale permet de calculer le vecteur d'état à tout instant sans recourir à la régularisation. Néanmoins le problème est alors de calculer une solution qui soit un bon compromis entre l'équation d'évolution et les observations disponibles. Ceci peut être réalisé en utilisant le formalisme de l'*assimilation variationnelle de données*. Cette technique permet non seulement la prise en compte de l'information temporelle mais encore de traiter naturellement le problème crucial des données manquantes. Nous présenterons alors nos contributions pour le calcul de l'estimation du mouvement.

Les modèles images étant généralement non linéaires, l'inversion de tels modèles est plus délicate numériquement (selon leur complexité). Nous distinguons dans l'état de l'art trois grandes approches.
La première approche consiste à considérer malgré tout la fonctionnelle de coût bâtie sur le modèle non linéaire, de calculer sa différentielle, puis de résoudre l'équation aux dérivées partielles (EDP) non linéaire qui en résulte. C'est ce qui est fait par exemple pour les contours actifs [**?**]. Elle requiert un fort coût algorithmique, car elle nécessite un algorithme itératif et pour chaque itération l'inversion d'une matrice de grande taille (selon la taille du vecteur d'état). Cela peut également mener à des instabilités numériques si la matrice est mal conditionnée ce qui est souvent le cas.
La seconde approche consiste à "linéariser" les modèles : il s'agit simplement d'approximer les modèles par un développement limité d'ordre 1. Le modèle étant linéaire, la différentiation de la fonctionnelle de coût conduit à résoudre un système d'EDP linéaires : la solution est obtenue

par discrétisation, ce qui conduit à résoudre un système linéaire de grande taille avec des méthodes itératives numériquement robustes. Une telle approche est séduisante mais possède évidemment un inconvénient : le modèle étant linéarisé, il est imparfait et la solution approximative. Pour le cas du flot optique, on peut s'en contenter si les vitesses à calculer sont faibles, ce qui n'est pas toujours le cas.
Enfin une voie intermédiaire consiste à utiliser des méthodes incrémentales. Le vecteur d'état est décomposé en un vecteur "moyen", que l'on suppose connu, et un vecteur incrémental à calculer. Le modèle est linéarisé et permet alors de calculer le vecteur incrémental à partir d'une solution imprécise. En ajoutant les deux vecteurs, on obtient une solution plus précise et le procédé peut alors être répété. Ainsi, on peut obtenir une solution avec la précision souhaitée mais le calcul reste néanmoins linéaire. Chacune de ces techniques a été utilisée à un moment ou à un autre dans ces travaux. La méthode incrémentale joue également un rôle important en assimilation variationnelle de données comme nous le verrons.

Le manuscrit comporte 5 chapitres. Le chapitre 1 récapitule les travaux effectués en imagerie biologique dans le contexte d'objets sphériques et illustre comment on peut exploiter une information spatiale pour résoudre des problèmes de traitement d'images. Le chapitre 2 présente brièvement l'assimilation variationnelle de données et résume nos contributions pour l'assimilation des images pour le calcul du flot optique. Enfin, le chapitre 3 décrit brièvement des recherches parallèles ou secondaires. Nous y traitons de méthodes pour le calcul du flot optique dans sa formulation non linéaire, des travaux effectués en occupation des sols par le biais de l'imagerie satellitaire et enfin de suivi multi-cibles dans des séquences d'images par l'utilisation de filtrage particulaire. Ce dernier est une approche probabiliste pour calculer une trajectoire selon un modèle d'évolution et en recoupant au mieux des observations partielles. Le chapitre 5 est un bref *curriculum vitæ* de l'auteur. Enfin le lecteur trouvera en annexe quelques publications en complément des chapitres précédents.

Chapitre 1

Prise en compte des informations spatiales : un cas d'étude

Dans ce chapitre sont présentées les contributions méthodologiques pour les problèmes de traitement d'images au sens général à partir de séquence d'acquisitions en microscopie *multi-focales*. Ce type de données est rencontré en imagerie biologique en optique focale (voir section 1.1) où la profondeur optique du dispositif d'acquisition varie. Une situation similaire est rencontrée en imagerie satellitaire où certains capteurs sont maintenant capables d'acquérir des images sur un très grand nombre de fréquences électromagnétiques. Il s'agit alors de données *multispectrales* et non plus de données multifocales. Les problèmes abordés sont l'estimation du mouvement, la reconstruction en volume et en temps (3D+t) à partir de données 3D statiques et d'une séquence d'images bidimensionnelles (2D+t), et le traitement d'images pour analyser des structures évoluant sur des surfaces sphériques. Ce choix est guidé par le contexte expérimental qui sous-tend ces travaux : nous étudions des séquences multifocales observant des simulations physiologiques de membranes cellulaires et ces membranes sont quasi-sphériques.

Nous ferons, selon les applications, l'une ou les deux hypothèses suivantes :

1. nous connaissons le processus d'acquisition d'une image à partir d'une scène 3D ;
2. nous observons des objets translucides quasi-sphériques sur lesquels évoluent des structures d'intérêt.

Les problèmes abordés sont par nature mal posés. De plus, nous ajoutons une difficulté supplémentaire puisque nous souhaitons produire un résultat 3D à partir de données qui ne le sont que partiellement. Les hypothèses faites constituent alors les informations *a priori* qui permettent de lever en partie l'indétermination. Cette indétermination est ensuite levée entièrement après régularisation spatiale des solutions.

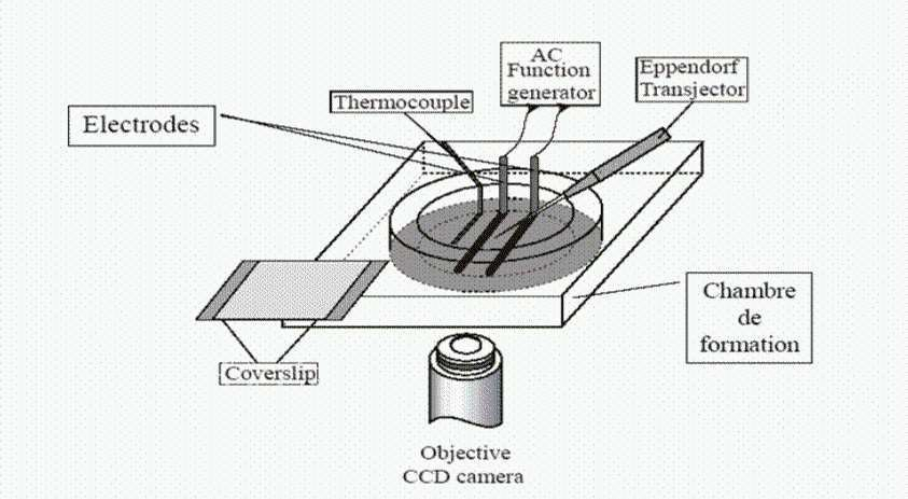
Dans la section 1.1 nous présentons plus en détail le contexte expérimental et la nature des données. Dans la section 1.2 nous présentons une première approche générique pour ana-

lyser la texture des structures sphériques présentes dans nos jeux d'images biologiques. Ces méthodes reposent sur la prise en compte de la géométrie de ces structures et donc de la seconde des hypothèses énumérées dans le paragraphe précédent. Dans la section 1.3, nous allons plus loin dans le traitement de ces données puisque il s'agit de reconstruire une séquence d'images 3D à partir d'une séquence d'images 2D et d'une initialisation 3D. Toutes ces méthodes utilisent la première des deux hypothèses précédemment énumérées et certaines d'entre elles font l'hypothèse de topologie sphérique. Enfin nous concluons à la section 1.4 et donnons une perspective scientifique à cette série de travaux.

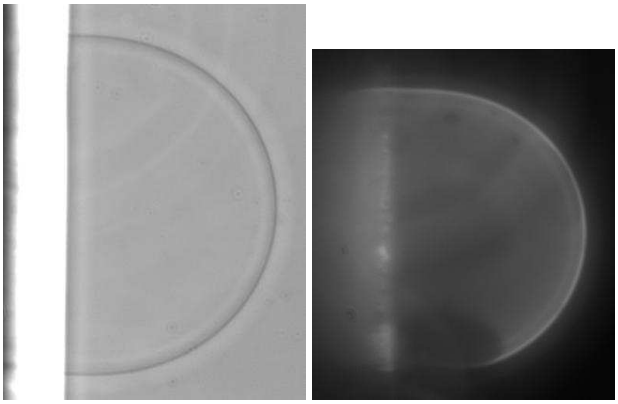
1.1 Contexte expérimental

Le contexte est donc, nous l'avons dit, l'imagerie biologique microscopique. Le dispositif d'acquisition est relativement simple et classique : il s'agit d'un microscope en optique focale couplé à un dispositif d'acquisition numérique (capteur CCD), voir figure 1.1(a). On remarquera aussi la présence d'électrodes et d'un micro injecteur dont l'utilité est expliquée dans le paragraphe suivant. Le microscope possède deux modes d'acquisition : l'acquisition en *contraste de phase* et l'acquisition en *fluorescence*. En contraste de phase, le photon en changeant de milieu modifie sa phase et c'est cette modification qui est détectée par le microscope. Elle permet la détection d'objets peu opaques. En fluorescence, on introduit du matériau fluorescent qui se fixe sur certaines molécules et qui sera ensuite détecté par le capteur. La figure 1.1(b) montre ces deux modalités. Les temps d'acquisition varient largement selon le mode de fonctionnement du microscope. En contraste de phase, le temps d'acquisition d'une image est de 20 millisecondes. Selon les marqueurs fluorescents utilisés et leur concentration, les temps d'acquisition varient de 100 millisecondes jusqu'à 4000 millisecondes. En mode contraste de phase, nous avons accès à une excellente précision temporelle mais les structures sont faiblement contrastées. En fluorescence, la résolution temporelle peut se révéler insuffisante pour étudier la dynamique de certaines structures mais le contraste des objets d'intérêts est bon.

La scène observée est en revanche moins classique. Les biologistes souhaitent étudier les échanges chimico-physiques de la membrane cellulaire avec divers agents. Pour cela, nos collègues biologistes créent des simulations de parois cellulaires appelées *vésicules* : ces structures sont sphéroïdes et constituées d'une double paroi phospholipidique comme le montre la figure 1.2(a). Les vésicules sont obtenues par hydratation de films lipidiques sous l'action d'un champ électrique : les parties *hydrophobes* des lipides, en fuyant l'eau, se regroupent et forment l'intérieur de la membrane tandis que les parties *hydrophiles* forment l'extérieur de la membrane. La différence de potentiel permet de faire croître la taille de la vésicule. Une fois la vésicule formée (cela peut prendre plusieurs heures), un manipulateur insère (à l'aide du micro manipulateur) des agents dans la membrane, formant une région appelée *raft* et étudie son comportement au cours du temps. La figure 1.2(b) montre une acquisition en fluorescence d'une vésicule et où un *raft* est bien visible.



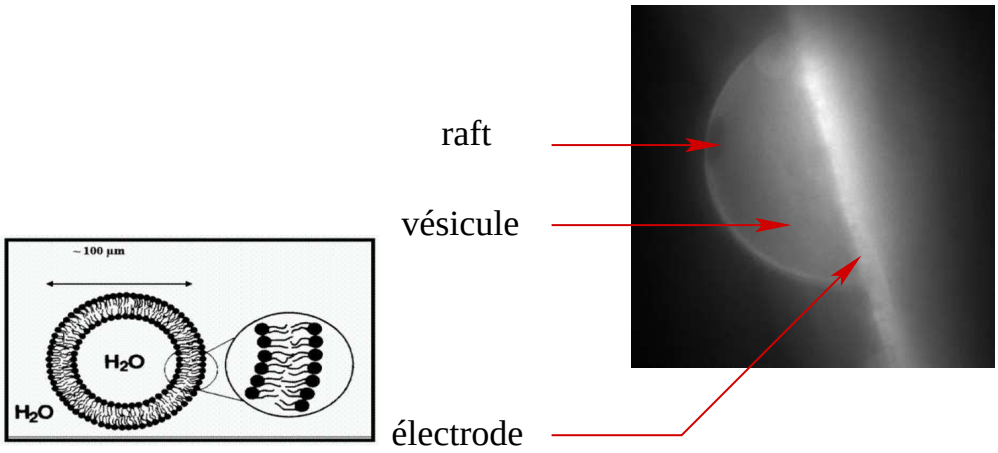
(a) Protocole expérimentale d'acquisition des données.



(b) Deux types d'acquisitions : contraste de phase (à gauche) ou fluorescence (à droite).

Figure 1.1 – Le dispositif d'acquisition des images.

Nous travaillons sur des données acquises en fluorescence car seule cette modalité permet la visualisation du *raft* sur la vésicule. Le microscope peut alors réaliser une série d'acquisitions, à une profondeur optique fixe, qui permet d'étudier l'évolution des objets. La figure 1.3 illustre ce type de séquence. Le logiciel d'acquisition d'image pilote également les paramètres de fonctionnement du microscope. On peut alors faire varier la profondeur optique du dispositif au fur et à mesure des acquisitions. On obtient ainsi une séquence dite multifocale ou *scan-f*. Comme le montre la figure 1.4, la séquence montre la vésicule à différentes profondeurs optique, on voit une vésicule nette dans la plus grande largeur mais de plus en plus floue à mesure qu'on avance ou que l'on recule. Il est en théorie possible d'obtenir une image nette à différentes profondeurs z et donc des images 3D mais cela nécessite une déconvolution ainsi qu'une phase de calibration pour estimer la fonction d'étalement du point (*PSF*). Pour des raisons techniques liées au contexte opératoire de formation des vésicules, il n'a pas été possible de mettre en œuvre ces



(a) Double membrane de phospholipides de la vésicule.

(b) Structures observées au microscope (fluorescence).

Figure 1.2 – Vésicules et rafts.

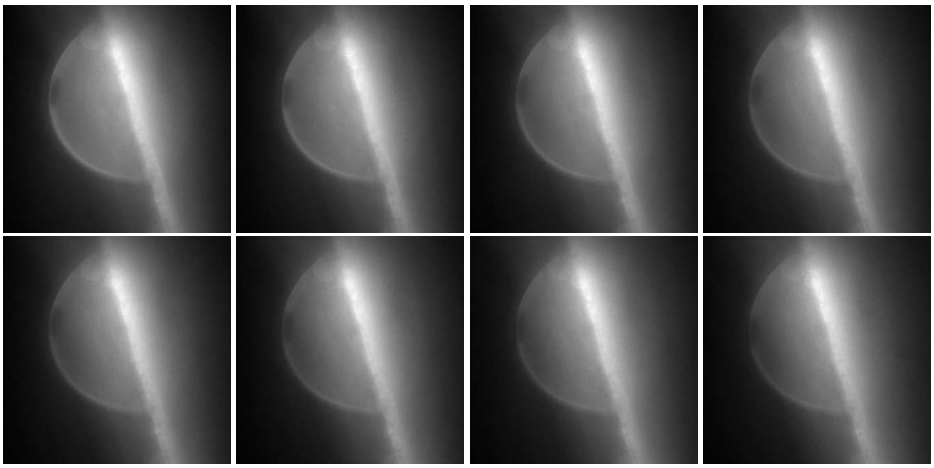


Figure 1.3 – Un exemple de séquence temporelle : on observe un déplacement du *raft*.

procédés de déconvolution. Les méthodes de reconstruction 3D décrites dans la section 1.3 sont donc appliquées à des données synthétiques en attendant d'obtenir de véritables données 3D. Néanmoins, nous proposons dans la section 1.4 une heuristique originale pour procéder à une reconstruction 3D sans phase de calibration.

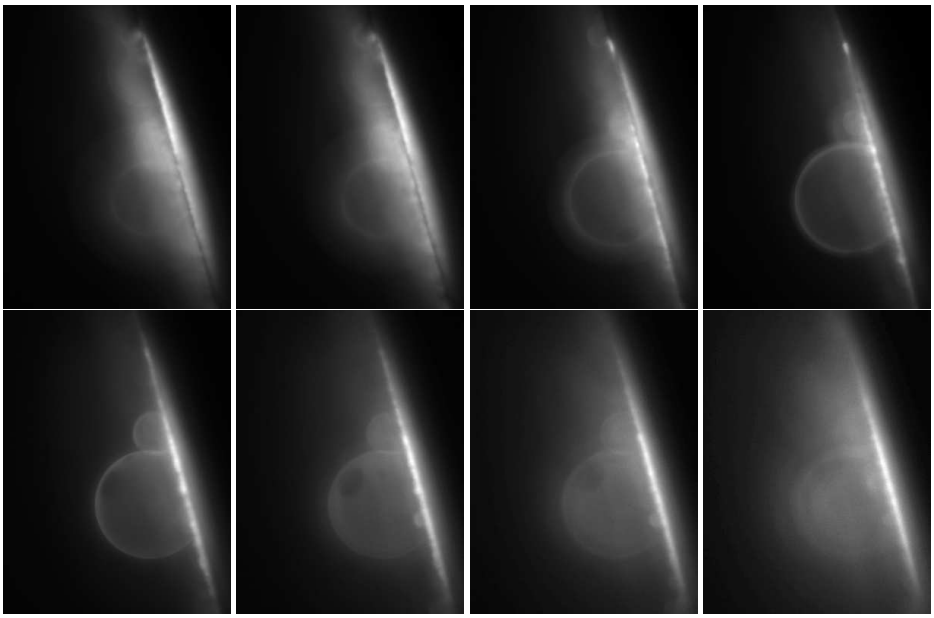


Figure 1.4 – Un exemple de séquence *scan-f* acquise à différentes profondeurs optique.

1.2 Traitement d'images dans un contexte sphérique

Dans cette section, nous tentons d'exploiter les informations *a priori* contenues dans les images de vésicule : la vésicule est translucide et quasi-sphérique ; et nous observons une projection cavalière de cette dernière dans l'axe de l'objectif du microscope. Pour traiter au mieux de ce type de données, nous proposons trois approches génériques.

La première méthode concerne les images 2D qui sont le résultat direct d'acquisition du microscope. Sur de telles images, il est aisé de déduire le rayon et le centre de la vésicule : appliquons un détecteur de contours puis un détecteur de Hough [**?**] et nous trouvons très facilement ces paramètres. Les coordonnées de la sphère-vésicule étant connues, nous sommes en mesure d'inverser le processus d'acquisition en plaquant l'image 2D de la vésicule sur la sphère 3D reconstruite. Seuls les objets (les *rafts*) qui évoluent sur la sphère nous intéressent, c'est-à-dire la texture de la vésicule. Comme nous souhaitons analyser ces structures (conformation spatiale, évolution dans le temps), il s'agit d'un problème 2D car les *rafts* sont partie intégrante de la membrane vésiculaire. L'idée est alors de transformer la surface de la sphère en un plan à l'aide d'une transformation adéquate. Pour ce faire, nous utiliserons les transformations cartographiques. Il existe un très grand nombre de ces transformations : par exemple un article du

XIX ième Siècle en recense plus de 200 ! En effet, une sphère n'est pas isomorphe à un plan, et il faut alors faire des compromis. Pour notre part, nous utilisons la projection de Lambert : elle permet de projeter un voisinage d'un point de la sphère sur le plan tangent à ce point et donc avec un minimum de déformation dans ce voisinage. Cette projection transforme un point M, identifié par la longitude λ et sa latitude φ dans le voisinage d'un point (un pôle) de référence P de longitude λ_P et latitude φ_P dans un système de coordonnées angulaire intermédiaire (z, a) par la formule :

$$\begin{cases} \cos(z) &= \sin(\varphi_P)\sin(\varphi) + \cos(\varphi_P)\cos(\varphi)\cos(\delta\lambda),\ \delta\lambda = \lambda_P - \lambda \\ \sin(a) &= \frac{\cos(\varphi)\sin(\delta\lambda)}{\sec(z)} \end{cases} \tag{1.1}$$

On obtient ensuite les coordonnées de la projection de Lambert qui sont en fait les coordonnées polaires dans le voisinage de P :

$$\begin{cases} r &= 2\sin(\frac{z}{2}) \\ \theta &= a \end{cases} \tag{1.2}$$

La figure 1.5 illustre les deux systèmes de coordonnées.

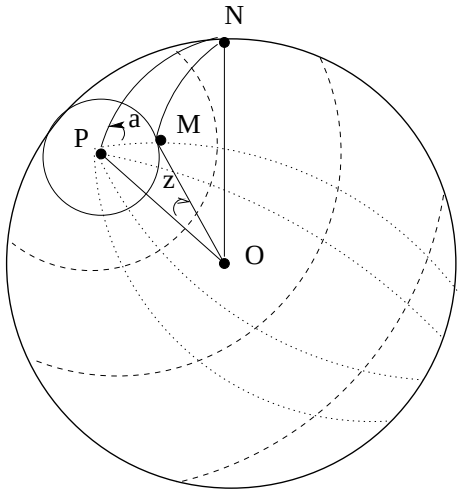


Figure 1.5 – Les coordonnées angulaires intermédiaires (z, a) d'un point M avec référence à un pôle P.

La seconde approche concerne les données 3D observant la vésicule. Elles doivent provenir des acquisitions multi-focales déconvoluées, et en l'état ce sont des données synthétiques. Là encore, en appliquant un détecteur de Hough, il est possible de déterminer le rayon de la sphère : il faut procéder plan par plan et ne garder que le rayon le plus grand et son centre associé. En appliquant la projection de Lambert, nous reconstruisons localement un plan de la vésicule. La figure 1.6 résume ce cas de figure ainsi que le précédent.

L'utilisation de projections cartographiques permet alors d'améliorer significativement le traitement des images 2D de vésicules puisque leur nature sphérique est prise en compte. Si

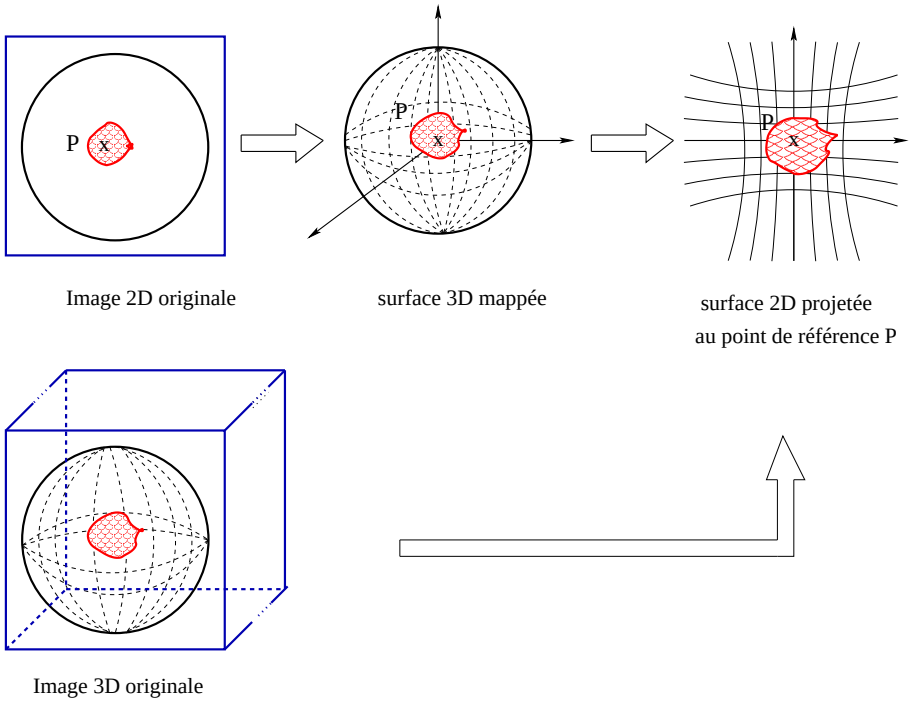


Figure 1.6 – Passage des données 2D et 3D aux références cartographiques.

les données sont 3D, elle permet de ramener un problème 3D à un problème 2D et donc une baisse significative du coût des calculs. Ce principe a été mis en œuvre pour concevoir un logiciel de visualisation d'informations reposant sur une sphère à partir des deux types de données décrites dans les deux paragraphes précédent. Ce logiciel, appelé MAPVIS [**?**] permet de visualiser des informations scalaires (la texture par exemple) ou bien vectorielles (un champ de vitesses par exemple). La projection d'une sphère sur un plan laisse apparaître un grand nombre de points non initialisés à cause des pas de discrétisation de l'image. Il est nécessaire d'interpoler ces points à partir des points voisins correctement initialisés. Nous utilisons une méthode d'interpolation multiéchelle sur une base de B-spline [**?**] afin d'obtenir un bon compromis en un résultat lisse et un résultat fidèle. La figure 1.7 montre un affichage de la texture tel que le permet MAPVIS et on mesure l'intérêt de cette projection pour l'application d'algorithme de traitement d'images 2D comme nous pouvons le voir dans la figure 1.8 : un algorithme de segmentation a été appliqué sur l'image brute et l'image projetée [**?**]. L'algorithme de segmentation est un contour actif qui n'utilise pas d'informations de type contours/gradient [**?**] pour se positionner sur les bords du *raft*, il donne de bons résultats sur ces images où les bords du *raft* sont peu marqués. Une autre application de ce principe a été un algorithme de suivi des *rafts* par une méthode de type *bloc-matching* sur un critère de maximum de corrélation. L'originalité de la méthode est l'utilisation de la projection de Lambert pour rectifier les deux blocs à mettre en correspondance [**?**, **?**, **?**] (voir annexe A.2 du manuscrit) et nous avons montré que seule la rectification par projection de Lambert permet de suivre tout le long de la séquence le

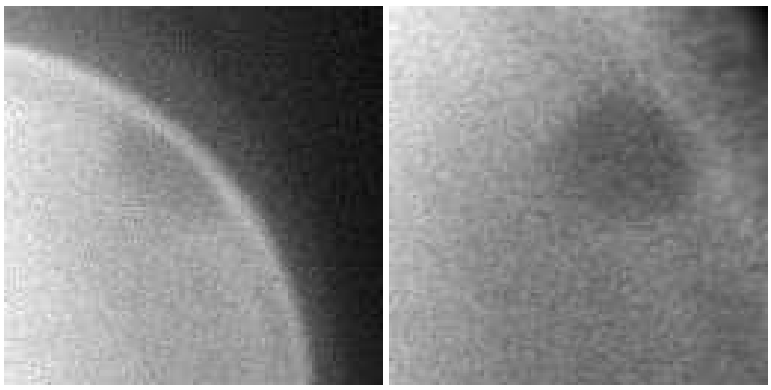


Figure 1.7 – Visualisation de la texture sans et avec MAPVIS. Le pôle a été choisi au centre du *raft*.

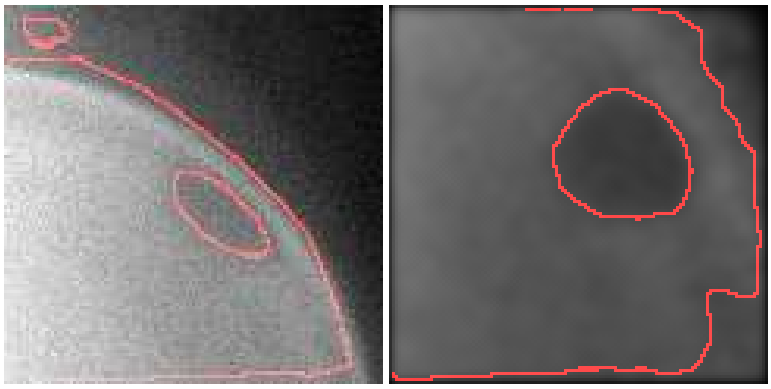
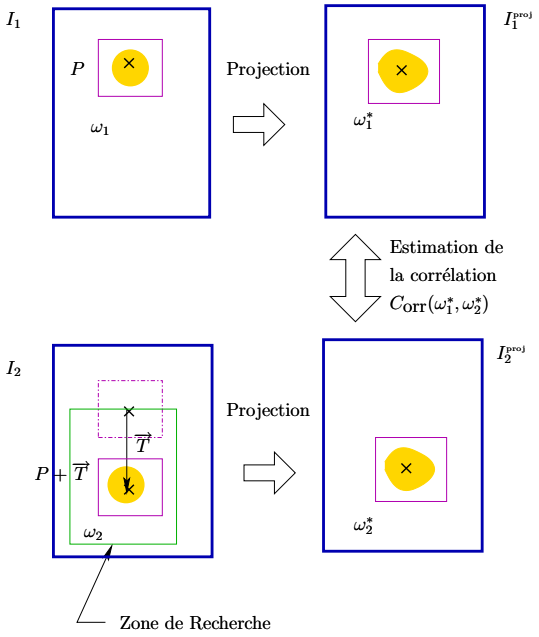
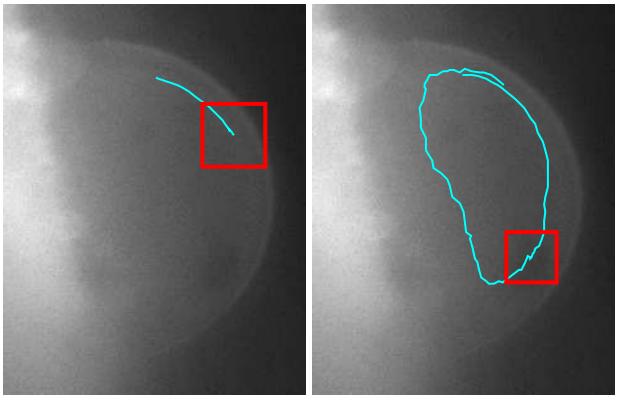


Figure 1.8 – Segmentation du raft sur image brute, puis projetée.

raft. La figure 1.9(a) résume l'algorithme proposé : le principe est de faire la recherche du bloc à mettre en correspondance dans l'image brute, mais le calcul de corrélation porte sur les images projetées dont chaque pôle est donné par les centres des blocs dans les images brutes. Sans la



(a) L'algorithme du *block-matching* projeté.



(b) Suivi d'un *raft* sur image brute et projetée. Le cadre rouge indique la fenêtre de calcul de la corrélation au plan 33, la ligne cyan est la trajectoire complète calculée par l'algorithme de suivi.

Figure 1.9 – Suivi par blocs projetés.

projection, l'algorithme de suivi du *raft* échoue très rapidement : le *raft* se rapproche du bord et le calcul de corrélation est biaisé par le bord de la vésicule (voir figure 1.9(b)). En revanche, en utilisant notre algorithme de suivi avec projection, nous réussissons à suivre le *raft* tout le long de la séquence longue d'une soixantaine de plans (voir figure 1.9(b)).

Enfin, une troisième approche pour prendre en compte la structure sphérique de la vésicule est de formuler le problème 3D en problème 2D par l'utilisation d'un paramétrage sphérique au lieu de procéder à la projection de Lambert. On traite donc directement la surface 3D plaquée (voir figure 1.6) ce qui permet d'éviter les problèmes d'interpolations liées à la projection de Lambert. Mais le problème doit alors être spécifiquement reformulé lorsque cela est possible. Nous avons fait ce travail pour le calcul du flot optique [**?**] (voir annexe A.2 de ce manuscrit). La contrainte du flot optique est reformulée en coordonnées sphériques (ici pour une sphère de rayon 1 et de centre C) :

$$I(\overrightarrow{CM}, t) = I(\overrightarrow{CM} + \partial\overrightarrow{CM}, t + \partial t), \tag{1.3}$$

où M est un point sur la sphère. On note $(\overrightarrow{e_r}, \overrightarrow{e_\theta}, \overrightarrow{e_\varphi})$ le repère local de Frenet au point M. Nous

avons alors :

$$\begin{aligned}\overrightarrow{CM} &= \overrightarrow{e_r} \\ \overrightarrow{e_r} &= \partial\theta\partial\overrightarrow{e_\theta} + \sin(\theta)\partial\varphi\partial\overrightarrow{e_\varphi}\end{aligned}$$

Nous reportons l'expression de $\overrightarrow{e_r}$ dans l'équation (1.3) et effectuons un développement de Taylor d'ordre 1 pour obtenir l'équation de contrainte sphérique du flot optique :

$$\frac{\partial I}{\partial\theta}\frac{\partial\theta}{\partial t} + \sin(\theta)\frac{\partial I}{\partial\varphi}\frac{\partial\varphi}{\partial t} + \frac{\partial I}{\partial t} = I_\theta\dot{\theta} + \sin\theta I_\varphi\dot{\varphi} + I_t = 0 \tag{1.4}$$

Cette équation est linéaire en $(\dot{\theta},\dot{\varphi})$ qui est le vecteur vitesse en coordonnées sphériques. Une solution unique est obtenue par régularisation spatiale en minimisant la fonctionnelle de coût :

$$\int((I_\theta\dot{\theta} + \sin\theta I_\varphi\dot{\varphi} + I_t)^2 + \alpha^2(\|\nabla\dot{\theta}\|^2 + \|\nabla\dot{\varphi}\|^2))\,d\dot{\theta}\,d\dot{\varphi} \tag{1.5}$$

Le paramètre α est un réel strictement positif choisit empiriquement qui pondère les effets de la régularisation. L'optimum est obtenu par un calcul de la variation de la fonctionnelle.

Dans cette section, nous avons donc utilisé la géométrique sphérique de la vésicule, tantôt pour aplanir la vésicule autour d'un pôle pour en faire une image 2D, tantôt pour plaquer des textures 2D et construire une pseudo image 3D. Dans ce dernier cas, nous plaquons toutes les structures visibles sur une vésicule sur une demi-sphère : en réalité la vésicule est transparente et les structures observées sur sa surface peuvent être à l'avant ou à l'arrière. Dans la section suivante, nous proposons des méthodes permettant de résoudre cette ambiguïté de positionnement.

1.3 Reconstruction 3D

Après avoir examiné les différentes possibilités pour faire du traitement d'images sur les données de vésicules, nous nous intéressons maintenant au problème de la reconstruction 3D à partir des séquences d'images 2D acquises par le microscope. Connaissant la taille de la vésicule et le dispositif d'acquisition, c'est-à-dire la façon d'obtenir une image 2D à partir de la scène 3D, il est possible de projeter la vésicule en vue planaire sur la structure sphérique 3D comme nous l'avons vu dans la section précédente. Néanmoins, la vésicule est translucide et les structures observées sur la vésicule ont un positionnement ambigu : sont-elles sur la face postérieure ou antérieure ? En se rappelant que la reconstruction 3D est possible (en théorie) à partir du mode d'acquisition multifocale du microscope, nous avons supposé que nous disposons de temps en temps de telles images : le microscope acquiert continuellement des images ce qui fournit une séquence 2D temporelle, puis on fait varier la profondeur optique le temps de balayer la vésicule en profondeur et on obtient, après déconvolution, une image 3D. La situation est donc telle que la montre la figure 1.10. Dans les acquisitions 3D, nous n'avons plus d'ambiguïté de positionnement des structures. Nous utilisons alors le fait que le décalage temporel entre une image 3D et l'acquisition 2D suivante est suffisamment faible pour lever

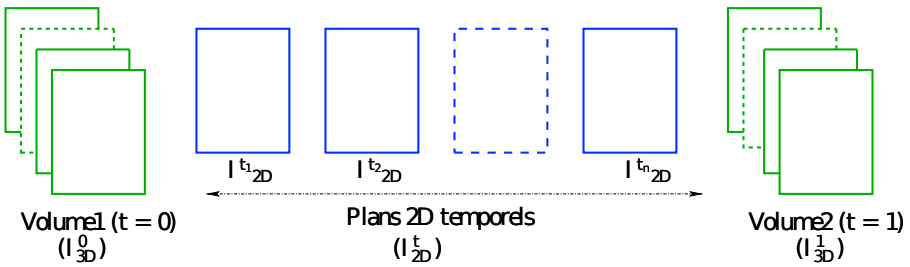


Figure 1.10 – Acquisition d'images 2D ou 3D

l'ambiguïté de positionnement dans la donnée 2D. De proche en proche, nous pouvons alors reconstruire la séquence 3D complète. Dans cette section nous proposons plusieurs méthodes de reconstruction 3D. Toutes utilisent le fait qu'on connaît le processus d'acquisition, c'est-à-dire qu'une image 2D, I_{2D}, acquise par le microscope au temps t est reliée à la scène 3D c'est-à-dire l'image, I, que l'on souhaite reconstruire par une relation du type

$$p(I)(x,y) = \int_{-\infty}^{+\infty} I(x,y,z)h(z)dz \tag{1.6}$$

qui décrit l'interaction des photons collectés par un pixel du capteur avec le milieu caractérisé ici par la fonction h. La fonction p est donc une projection qui transforme une image 3D en une image 2D. Ceci nous donne une équation reliant l'observation I_{2D} à l'image I à reconstruire :

$$p(I)(x,y) - I_{2D}(x,y) = 0 \tag{1.7}$$

Encore une fois, on peut exploiter la propriété sphérique de la vésicule pour définir plus précisément la fonction p : dans le cas où l'on observe une seule vésicule, chaque pixel de l'image acquise contient les contributions de l'interaction des photons avec les parois avant et arrière de la vésicule comme le montre la figure 1.11. Nous proposons de reconstruire non pas une

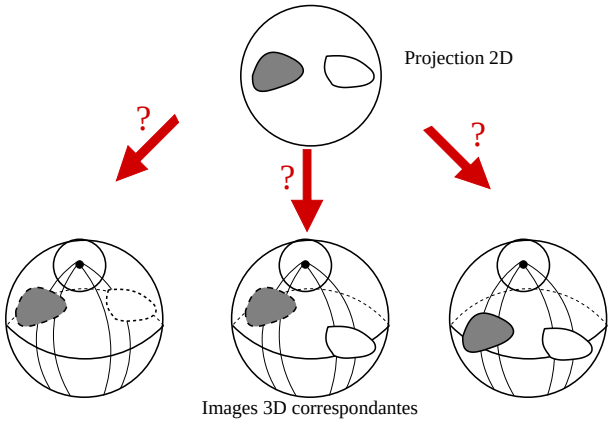


Figure 1.11 – Ambiguïté de positionnement.

demi-sphère comme nous l'avons fait dans la section précédente, mais maintenant une sphère

complète en dupliquant les deux hémisphères. On obtient alors une première reconstruction **imprécise** (voir figure 1.12) que l'on nomme I_G. L'équation de transport de la luminosité entre

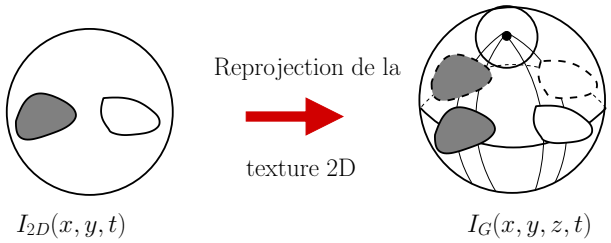


Figure 1.12 – Reconstruction imprécise.

deux acquisitions 3D s'écrit :

$$I(\mathbf{x},t) = I(\mathbf{x}-\mathbf{w},t-1) \tag{1.8}$$

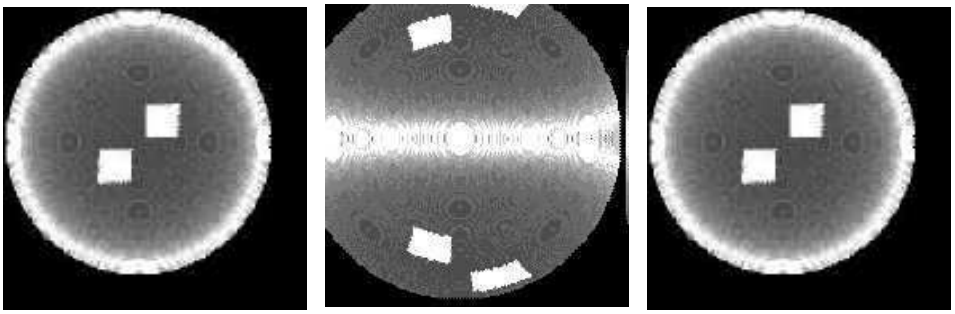
où $\mathbf{x} = (x,y,z)$ et $\mathbf{w}$ est un vecteur vitesse de $\mathbb{R}^3$. Nous avons exprimé ce transport sous forme rétrograde car il sera plus aisé par la suite de reconstruire l'image à partir du temps précédent sans avoir à faire d'interpolation. En linéarisant cette équation, on obtient une contrainte 3D du flot optique :

$$\nabla I(\mathbf{x},t-1)^T\mathbf{w} + I(\mathbf{x},t) - I(\mathbf{x},t-1) = 0 \tag{1.9}$$

où $\nabla I = (\frac{\partial I}{\partial x} \quad \frac{\partial I}{\partial x})^T$ est le gradient de I. Au temps $t-1$, l'image 3D, I, est connue, soit comme donnée initiale, soit comme reconstruction. Au temps t, I n'est pas encore reconstruite, nous utilisons en lieu et place la reconstruction imprécise I_G déduite de l'image I_{2D} correspondante. Nous utilisons alors la régularisation spatiale pour calculer un champs de vitesse régulier ce qui permet de lever l'ambiguïté de positionnement. Nous minimisons donc la fonctionnelle d'énergie :

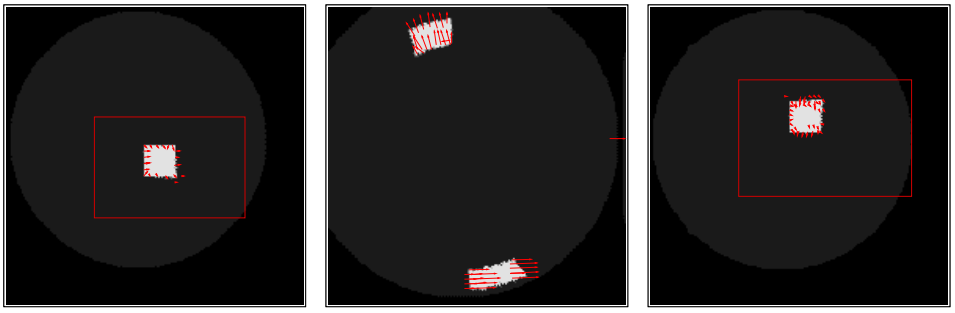
$$\int \left((\nabla I(\mathbf{x},t-1)^T\mathbf{w} + I_G(\mathbf{x},t) - I(\mathbf{x},t-1))^2 + \alpha^2 \|\nabla \mathbf{w}\|^2 \right) d\mathbf{x} \tag{1.10}$$

Le minimum est déterminé en utilisant les techniques du calcul des variations. La méthode a été validée sur des données synthétiques puisque les véritables données 3D en contraste de phase ne sont pas encore accessibles. Dans ces données, on construit deux carrés évoluant avec un mouvement propre et approximativement diamétralement opposé sur une sphère translucide. Nous utilisons une donnée initiale 3D et une séquence 2D en vue projective de cette sphère et calculons le champs de déplacement par minimisation de (1.10). La figure 1.13 montre la donnée 3D imprécise calculée à partir de la donnée 2D sous deux angles de vue avec MAPVIS, on remarquera que les deux carrés sont visibles. La figure 1.14 montre le champ de vitesses calculé et superposé à l'image 3D au temps précédent, la visualisation est toujours assurée par MAPVIS, l'ambiguïté a été levée. Ces résultats ont été publiés dans [**?**], consultable dans l'annexe A.3 de ce manuscrit. On trouvera d'avantage de résultats dans [**?**], chapitre 4.



(a) Vue frontale. (b) Vue au pôle Nord. (c) Vue dorsale.

Figure 1.13 – Vues projectives de la reconstruction imprécise.



(a) Vue frontale (b) Vue au pôle Nord (c) Vue dorsale

Figure 1.14 – Vues projectives du champ des vitesses calculées.

Nous avons proposé une seconde approche pour la reconstruction 3D+t toujours en exploitant la sphéricité de la vésicule en utilisant une démarche similaire à celle vue dans la section précédente utilisant le paramétrage sphérique pour se ramener à un problème 2D. Nous formulons d'abord la contrainte 3D du flot optique, équation (1.8), en coordonnées sphériques et en appliquant un développement de Taylor d'ordre 1 :

$$I(\theta,\varphi,t) = I_{t-1}(\theta,\varphi,t-1) + \frac{\partial I(\theta,\varphi,t-1)}{\partial\theta}\frac{\partial\theta}{\partial t} + \sin(\theta)\frac{\partial I(\theta,\varphi,t-1)}{\partial\varphi}\frac{\partial\varphi}{\partial t} \tag{1.11}$$

Nous faisons la même hypothèse sur p qu'au début de cette section : la vésicule est transparente et on observe dans l'image acquise des contributions venant des faces dorsale et frontale de la vésicule. Nous l'exprimons ainsi :

$$p(I)(\theta,\varphi) = I(\theta,\varphi) + I(\theta,\pi-\varphi) \tag{1.12}$$

Nous appliquons p aux deux membres de l'équation (1.11) et obtenons après simplification :

$$\begin{aligned} p(I_\theta)(\theta,\varphi,t)\dot{\theta}(\theta,\varphi) + p(I_\theta)(\theta,\pi-\varphi,t-1)\dot{\theta}(\theta,\pi-\varphi) + \\ p(I_\varphi)(\theta,\varphi,t-1)\dot{\varphi}(\theta,\varphi)\sin\theta + p(I_\varphi)(\theta,\pi-\varphi,t-1)\dot{\varphi}(\theta,\pi-\varphi) + \\ p(I)(\theta,\varphi,t-1) - I_{2D}(\theta,\varphi,t) = 0 \end{aligned} \tag{1.13}$$

Nous avons 4 inconnues : $(\dot{\theta}(\theta,\varphi),\dot{\varphi}(\theta,\varphi))$ le vecteur vitesse sur la face frontale et $(\dot{\theta}(\theta,\pi-\varphi),\dot{\varphi}(\theta,\pi-\varphi))$ pour la face dorsale. Cette sous-détermination provient de la projection p, nous ramenons le problème à deux inconnues en distinguant les objets appartenant aux deux hémisphères en supposant :

$$\begin{aligned}\dot{\theta}(\theta,\pi-\varphi) &= \dot{\theta}(\theta,\varphi)\\ \dot{\varphi}(\theta,\pi-\varphi) &= -\dot{\varphi}(\theta,\varphi)\end{aligned}$$

L'équation (1.13) est donc simplifiée et nous calculons une solution par régularisation spatiale. On consultera [**?**], donné dans l'annexe A.3, pour plus de détails.

Finalement, il est aussi possible de procéder à la reconstruction 3D+t sans tenir compte de la topologie de la vésicule. Ceci est notre dernière contribution [**?**], voir annexe A.3, à la reconstruction 3D. Nous utilisons bien sûr la connaissance de p. En considérant l'équation (1.9), nous utilisons la linéarité de p pour obtenir une équation 2D de flot optique :

$$p(\nabla I(\mathbf{x},t-1)^{\top}\mathbf{w}) + I_{2D}(x,y,t) - p(I(\mathbf{x},t-1)) = 0 \tag{1.14}$$

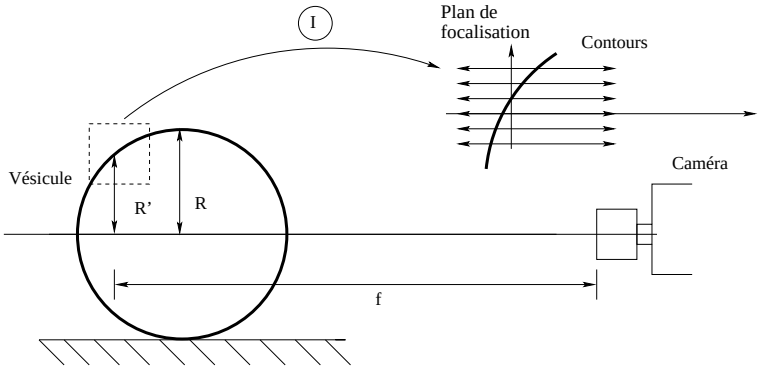
Une solution peut être obtenue par régularisation spatiale. Pour les deux méthodes, il faut bien évidemment connaître p qui doit être différentiable, ce qui est le cas si elle s'écrit comme dans (1.6). Pour montrer que la méthode fonctionne, nous l'avons mise en œuvre sur des données synthétiques et nous avons supposé une contribution égale à chaque profondeur optique, c'est-à-dire $p(I)(x,y) = \int_a^b I(x,y,z)\frac{dz}{b-a}$ puisque c'est ainsi qu'ont été construite les données. Dans le cas des données biologiques en contraste de phase ou fluorescence, la méthode est en l'état inapplicable puisque non seulement nous ne disposons pas des données 3D défocalisées mais encore la fonction p n'est pas connue.

1.4 Conclusions et perspectives

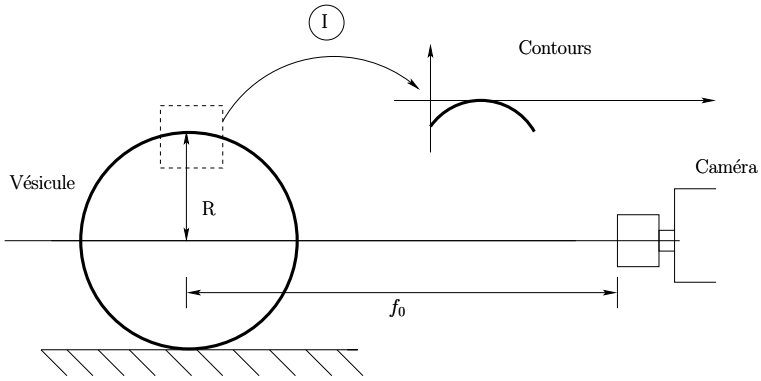
Nous avons présenté nos contributions sur le traitement d'images biologiques de vésicule acquises par un microscope focal standard. Ces travaux ont été réalisés dans le cadre de la thèse de Wafa Rekik [**?**] dont j'ai assuré la complète direction scientifique. Les deux dernières contributions, concernant la reconstruction 3D, n'ont pu être entièrement publiées autrement que par des rapports de recherche, essentiellement par manque de temps, et aussi à cause de mon départ en délégation.

Pour mesurer l'intérêt de ces méthodes, les tester sur des données synthétiques n'est pas suffisant, il faut pouvoir les mettre en œuvre sur des données réelles. Comme il a été dit au début de ce chapitre, le dispositif d'acquisition auquel nous avions un accès (indirect) ne pouvait pas restituer des images déconvoluées 3D. Aussi, en suite logique à ces travaux, il faut résoudre ce problème de déconvolution. À cette fin, j'ai proposé une stratégie possible et originale. Il s'agit bien d'un problème de déconvolution, puisqu'il s'agit de calculer I connaissant des relations du type $\int I(x,y,z)h(x,y,z,f)dz = I_{2D}(x,y,f)$ La fonction h dépend des conditions d'acquisition, en particulier de la profondeur optique f et $I_{2D}(x,y,f)$ est l'image acquise par le microscope. En examinant les données *scan-f*, telles que celles montrées dans la figure 1.4, on

observe que la vésicule est nette dans sa plus grande largeur. Au fur et à mesure qu'on augmente ou diminue la profondeur optique, la vésicule observée dans la zone de focalisation devient de plus en plus floue. La raison en est que les parties de la vésicule qui sont tangentes aux trajectoires des photons collectés par le capteur interagissent plus fortement (sur une longueur plus grande) comme le montre la figure 1.15. Nous pouvons exploiter ce constat pour définir une



(a) Image floue.



(b) Image nette au voisinage de l'équateur.

Figure 1.15 – Action de la profondeur optique sur la netteté.

fonction h adéquate. Soit z_f la profondeur qui focalise au mieux pour une focale f, la fonction h pourrait s'écrire :

$$h(x,y,z,f) = h(x,y)\frac{\exp(-\frac{z-z_f}{2\sigma_f})}{\sqrt{2\pi}\sigma_f}$$

en supposant un modèle de flou gaussien en profondeur. On pose $h(x,y) = 1$ par soucis de simplification. Les variances σ_f et les biais z_f restent à calibrer. On connaît déjà la focale f qui donne une image de la vésicule la plus nette. Connaissant le diamètre et le centre de la vésicule, il est facile alors de construire une image idéale binaire qui positionne les contours de la vésicule pour chaque valeur de z. Cette pseudo image reconstruite sert alors de mire de calibration pour déduire les paramètres z_f et σ_f en appliquant les méthodes de déconvolution de l'état de l'art. Par la suite, la modélisation de h pourra être affinée en utilisant des fonctions d'étalement du

point plus réalistes qu'un simple flou en z. Ce thème de recherche sera proposé comme futur sujet de thèse.

Chapitre 2

Prise en compte de la dynamique : assimilation variationnelle de données

Le but de ce chapitre est de montrer comment on peut incorporer la connaissance de la dynamique d'une séquence d'images dans la formulation variationnelle d'un problème de traitement d'images. Ces considérations sont générales et indépendantes de la dimension du problème même si nous utilisons le problème de la détermination du flot optique comme exemple d'application. Nous supposons donc que nous disposons d'une séquence d'images et que nous connaissons la dynamique de la quantité que l'on souhaite calculer à partir de l'observation de cette séquence d'images.

Dans ce chapitre, nous notons $\mathbf{Y}$ la séquence spatio-temporelle des observations et $\mathbf{X}$ le vecteur d'état. Le vecteur d'état désigne la ou les grandeurs physiques (par exemple la température, la radiance ou encore le vecteur vitesse) que l'on souhaite déterminer à partir des observations et des lois qui régissent le comportement du vecteur d'état. Connaître la dynamique du vecteur d'état, c'est disposer d'une équation d'évolution généralement écrite sous la forme :

$$\frac{\partial \mathbf{X}}{\partial t}(\mathbf{x},t) + M(\mathbf{X}(\mathbf{x},t)) = \mathcal{E}_Q(\mathbf{x},t) \tag{2.1}$$

L'opérateur M est appelé modèle d'évolution, $\mathbf{x}$ est une coordonnée spatiale et t le temps. Pour intégrer l'équation (2.1), nous devons disposer d'une condition initiale sur $\mathbf{X}$:

$$\mathbf{X}(\mathbf{x},t_0) = \mathbf{X}_0(t) + \mathcal{E}_B(\mathbf{x}) \tag{2.2}$$

Nous disposons également d'une relation entre les observations et le vecteur d'état. On peut la trouver sous cette forme dans la littérature

$$\mathbf{Y}(\mathbf{x},t) = H(\mathbf{X}(\mathbf{x},t)) + \mathcal{E}_R(\mathbf{x},t) \tag{2.3}$$

mais nous la généralisons à :

$$H(\mathbf{X}(\mathbf{x},t),\mathbf{Y}(\mathbf{x},t)) = \mathcal{E}_R(\mathbf{x},t) \tag{2.4}$$

ce qui nous permettra de formuler correctement le problème de l'estimation du flot optique comme nous le verrons. L'opérateur H est appelé opérateur d'observation. Les vecteurs $\mathcal{E}_Q$, $\mathcal{E}_B$ et $\mathcal{E}_R$ représentent respectivement les erreurs du modèle d'évolution, du terme d'ébauche et du modèle d'observation. Pour le moment, nous ne faisons pas d'hypothèse particulière concernant ces bruits. Le problème est donc de trouver une solution $\mathbf{X}$ qui satisfasse au mieux les équations (2.1), (2.2) et (2.3).

Nous pouvons formuler la résolution du problème avec l'approche variationnelle classiquement employée en traitement d'image, c'est-à-dire minimiser la fonctionnelle de coût suivante :

$$J(\mathbf{X}) = \int_\Omega \int_{t_0}^{t_1} (\frac{\partial \mathbf{X}}{\partial t} + M(\mathbf{X}))^2 d\mathbf{x}dt + \int_\Omega \int_{t_0}^{t_1} H(\mathbf{X},\mathbf{Y})^2 d\mathbf{x}dt + \int_\Omega (\mathbf{X}-\mathbf{X}_0)^2 d\mathbf{x} \tag{2.5}$$

Contrairement aux approches vues dans le chapitre précédent, la solution est recherchée maintenant dans l'espace spatio-temporel $\Omega \times [t_0, t_1]$, l'intervalle $[t_0, t_1]$ étant le domaine temporel. Notons que pour le moment, nous ne prenons pas en compte la nature des erreurs des modèles.

Appliquons ceci au problème de l'estimation du flot optique. Le vecteur d'état est donc le champ de vecteurs vitesse dans l'image. Nous avons alors $\mathbf{X}(\mathbf{x},t) = (u(\mathbf{x},t), v(\mathbf{x},t))^T$ où u et v sont les composantes horizontale et verticale du vecteur vitesse. Nous avons besoin d'un modèle d'évolution. Le choix de ce modèle dépendra des conditions expérimentales. Dans le cas de l'estimation de la vitesse d'objets dans une séquence vidéo quelconque, nous ne disposons d'aucune information sur la dynamique. Nous ferons alors une hypothèse qui, même très simple et limitative, nous permettra d'appréhender l'apport de l'information temporelle dans le traitement d'un problème. Cette hypothèse est la constance d'un vecteur vitesse le long de sa trajectoire. Elle se traduit par :

$$\frac{d\mathbf{X}}{dt} = 0 \tag{2.6}$$

$$\frac{\partial \mathbf{X}}{\partial t} + \nabla \mathbf{X}^T \mathbf{X} = 0 \tag{2.7}$$

L'équation (2.7) se réécrit sous forme scalaire :

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}u + \frac{\partial u}{\partial y}v = 0 \tag{2.8}$$

$$\frac{\partial v}{\partial t} + \frac{\partial v}{\partial x}u + \frac{\partial v}{\partial y}v = 0 \tag{2.9}$$

L'équation d'observation lie les observations au vecteur d'état, nous prenons donc naturellement l'équation du flot optique :

$$\nabla I^T \mathbf{X} + \frac{\partial I}{\partial t} = 0 \tag{2.10}$$

On remarque que la généralisation de l'équation d'observation (équation (2.4)) était nécessaire dans le cas de l'estimation du flot optique. Par souci de lisibilité, nous utilisons la notation in-

dice pour les dérivées partielles, par exemple $\frac{\partial u}{\partial t} = u_t$. Finalement la fonctionnelle de coût devient :

$$\begin{aligned} J(u,v) &= \int_\Omega \int_{t_0}^{t_1} [(u_t + u_x u + u_y v)^2 + (v_t + v_x u + v_y v)^2 + (I_x u + I_y v + I_t)]^2 \, dx dy dt \\ &+ \int_\Omega (u(t=t_0) - u_0))^2 dx dy + \int_\Omega (v(t=t_0) - v_0)^2 dx dy \end{aligned}$$

Calculons la différentielle des deux premiers termes intégrants de J (que nous nommons J_1) par rapport à u :

$$\begin{aligned} \frac{\partial J_1}{\partial u} = 2 \quad & [-u_{tt} - u_{xx} u^2 - u_x^2 u - 2 v v_y u_y \\ & - v^2 u_{yy} - 2 u_{tx} u - u_x u_t - v_t u_y - 2 v u_{yt} - v_y u_t \\ & - v_x u u_y - v u u_{yx} - v_y u_x u_y - v u_{xy} u_y - v u_x u_{yy}] \end{aligned}$$

Sans développer davantage le calcul de la différentielle complète de J on remarque que l'on obtient une expression complexe du second ordre avec des termes croisés et non linéaires et cela malgré une équation d'évolution très simple. Une telle équation sera difficile à discrétiser et numériquement instable. Pour résoudre cette difficulté nous allons utiliser les algorithmes de type "4DVar" utilisés en assimilation de données.

2.1 Assimilation variationnelle de données

L'assimilation de données désigne les méthodes qui permettent de corriger, à l'aide d'observations, les prédictions d'un modèle physique. Il s'agit de bâtir une fonctionnelle de coût dont l'optimal fournit la solution. Ces méthodes ont été initialement appliquées aux sciences de l'environnement, c'est-à-dire la prévision météorologique, les modèles océanographiques, les modèles de qualité de l'air (polluants chimiques ou autres). Son utilisation commence à se généraliser à d'autres domaines : citons par exemple la biologie végétale pour la croissance de plantes, la médecine avec le projet CardioSense3D[1] de modélisation électromécanique du cœur. Enfin, signalons que son utilisation en traitement d'images reste pour l'heure encore émergente. L'assimilation de données étant un problème d'optimisation, nous nous sommes naturellement intéressé aux déclinaisons variationnelles de ces méthodes.

Cas pour un modèle d'évolution parfait

L'algorithme 4DVar (pour assimilation variationnelle spatio-temporelle) a été introduit au début des années 80 par [**?**]. Notons que c'est ce type d'algorithme qui est utilisé de manière opérationnelle chez Météo France pour les prévisions météorologiques. Si nous supposons que le modèle d'évolution est parfait, c'est-à-dire que $Q \equiv 0$; alors la donnée du vecteur d'état à l'instant initial suffit alors à connaître le vecteur d'état à tout instant. On suppose que l'on dispose

[1] http ://www-sop.inria.fr/CardioSense3D/

d'observations, qui peuvent être dégradées, du vecteur d'état à différents instants. Le problème se résume donc à trouver le terme d'ébauche, c'est-à-dire le vecteur d'état au premier instant, de façon telle qu'il soit le plus proche des observations. Il s'agit donc de minimiser la fonctionnelle suivante :

$$J(\mathbf{X}(t_1)) = \sum_{i=1,}^{n} < \mathbf{Y}_i - \mathbf{X}(t_i) | \mathbf{Y}_i - \mathbf{X}(t_i) > \tag{2.11}$$

$\mathbf{Y}_i$ est l'observation du vecteur d'état à l'instant t_i et $< | >$ est le produit scalaire à associé à l'espace fonctionnel de représentation des observations et du vecteur d'état. La minimisation de J sera obtenue par une descente de gradient. Nous calculons la différentielle de J : une variation $\delta\mathbf{X}(t_1)$ sur le vecteur d'état engendre la variation sur J :

$$\delta J = 2 \sum_{i} < \mathbf{X}_i(t_i) - \mathbf{Y}_i | \delta\mathbf{X}(t_i) > \tag{2.12}$$

Si nous écrivons l'équation d'évolution de $\mathbf{X}$ sous la forme suivante :

$$\mathbf{X}_t = M(\mathbf{X}) \tag{2.13}$$

nous en déduisons par différentiation l'équation d'évolution sur la variation de $\mathbf{X}$:

$$\delta\mathbf{X}_t = A(t)\delta\mathbf{X}(t) \tag{2.14}$$

où A est le jacobien de M. En intégrant cette équation, on en déduit l'opérateur résolvant R associé à (2.14) :

$$\delta\mathbf{X}(t) = R(t, t_1)\delta\mathbf{X}(t_1) \tag{2.15}$$

Nous pouvons maintenant faire apparaître $\mathbf{X}(t_1)$ dans le membre de droite de l'équation (2.12) :

$$\delta J = 2 \sum_{i} < \mathbf{X}(t_i) - \mathbf{Y}_i | R(t_i, t_1)\delta\mathbf{X}(t_1) > \tag{2.16}$$

Introduisons maintenant l'opérateur adjoint de R, noté R^*, et défini par :

$$< \mathbf{X}' | R\mathbf{X} > = < R^*\mathbf{X}' | \mathbf{X} > \tag{2.17}$$

Remarquons que selon la nature de R ou du produit scalaire, R^* n'est pas nécessairement la transposée de R. L'équation (2.16) devient :

$$\delta J = < 2 \sum_{i} R^*(t_i, t_1)[\mathbf{X}(t_i) - \mathbf{Y}_i] | \delta\mathbf{X}(t_1) > \tag{2.18}$$

Et l'opérande gauche du produit scalaire est par définition la différentielle de J. Nous allons maintenant montrer une méthode efficace pour évaluer l'opérande gauche du produit scalaire de (2.18) donc pour calculer la différentielle de J. L'équation (2.14) peut s'écrire sous la forme d'un opérateur linéaire appliqué à $\delta\mathbf{X}$ (puisque A est un opérateur linéaire et possède un adjoint) :

$$[\frac{\partial}{\partial t} - A(t)]\,\delta\mathbf{X}(t) = 0$$

Cet opérateur possède donc un adjoint défini par :

$$-\frac{\partial}{\partial t} - A^*(t)$$

Constatons que le signe - apparaît conséquemment à l'intégration par parties nécessaire pour le calcul de l'adjoint. On en déduit l'*équation adjointe* :

$$(\delta^*\mathbf{X})_t = -A^*(t)\delta^*\mathbf{X}(t) \tag{2.19}$$

On peut montrer que pour tout couple $(\mathbf{X},\mathbf{Z})$ respectivement solution de (2.14) et (2.19) on a $\frac{\partial}{\partial t} < \mathbf{X}(t)|\mathbf{Z}(t) >= 0$ et donc que $< \mathbf{X}(t)|\mathbf{Z}(t) >$ ne dépend pas de t. Soit S le résolvant de (2.19). Nous avons donc :

$$\begin{aligned} < \mathbf{X}(t_1)|\mathbf{Z}(t_1) > &= < \mathbf{X}(t_i)|\mathbf{Z}(t_i) > \\ < \mathbf{X}(t_1)|S(t_1,t_i)\mathbf{Z}(t_i) > &= < R(t_i,t_1)\mathbf{X}(t_1)|\mathbf{Z}(t_i) > \end{aligned}$$

Donc $R^* = S$, où S est le résolvant de l'équation adjointe (2.19) entre t_i et t_1. Intégrer (2.19) revient donc à calculer R^* et donc on peut calculer $\sum_i R^*(t_i,t_1)(\mathbf{X}(t_i)-\mathbf{Y}_i)$ c'est-à-dire $\frac{1}{2}\nabla J$. L'intégration de (2.19) s'obtient rétrogradement à partir de $X(t_n)$. Il faut donc d'abord calculer $X(t_n)$ à partir de $X(t_1)$ et de l'équation (2.13). Nous obtenons alors un algorithme efficace pour calculer le gradient de J et le minimisant de J à travers un méthode de descente de gradient. Cet algorithme est en deux passes :

1. **direct run** : à partir d'une condition initiale $\mathbf{X}(t_1)$ et l'équation (2.13) calculer $\mathbf{X}(t_2),\cdots,\mathbf{X}(t_n)$.
2. **backward run** : à partir de $\mathbf{X}(t_n)$ et l'équation (2.19) calculer $R^*(t_n,t_1),\cdots,R^*(t_2,t_1)$ puis ∇J.

Cette méthode est bien adaptée dans les cas où le modèle d'évolution décrit parfaitement la dynamique de l'image. C'est pour cette raison que cet algorithme est utilisé en météorologie, par exemple, car les phénomènes physiques observés sont relativement bien connus et modélisés. Mais dans le cas où le modèle d'évolution est imparfait, approximatif, cette approche est inadaptée car elle privilégie le modèle sur les observations.

Assimilation de données et modèle imparfait

Nous nous plaçons dans le cas où la dynamique du vecteur d'état est mal connue, comme c'est souvent le cas en traitement d'images. Nous souhaitons alors calculer un vecteur d'état qui soit un compromis entre l'adéquation au modèle et aux observations, c'est-à-dire considérer une énergie du type (2.5). Dans le cas général, le problème se formalise ainsi :

$$\begin{aligned} J(\mathbf{X}) = &\frac{1}{2}\int_A\int_A \left(\frac{\partial \mathbf{X}}{\partial t} + M(\mathbf{X})\right)^T(\mathbf{x},t)Q^{-1}(\mathbf{x},t,\mathbf{x}',t')\left(\frac{\partial \mathbf{X}}{\partial t} + M(\mathbf{X})\right)(\mathbf{x}',t')\,d\mathbf{x}dt\,d\mathbf{x}'dt' \\ &+\frac{1}{2}\int_A\int_A H(\mathbf{X},\mathbf{Y})^T(\mathbf{x},t)R^{-1}(\mathbf{x},t,\mathbf{x}',t')H(\mathbf{X},\mathbf{Y})(\mathbf{x}',t')\,d\mathbf{x}dt\,d\mathbf{x}'dt' \\ &+\frac{1}{2}\int_\Omega\int_\Omega (\mathbf{X}(\mathbf{x},0)-\mathbf{X}_0(\mathbf{x}))^T B^{-1}(\mathbf{x},\mathbf{x}')(\mathbf{X}(\mathbf{x}',0)-\mathbf{X}_0(\mathbf{x}'))\,d\mathbf{x}d\mathbf{x}' \end{aligned} \tag{2.20}$$

avec $A = \Omega \times [t_0, t_1]$. Nous faisons apparaître les matrices de covariances associées aux erreurs modèles et ainsi prenons en compte la nature de l'erreur. Nous supposerons que ce sont des bruits gaussiens de matrices de covariance respectives Q, B et R. On remarquera alors que la solution minimisant la fonctionnelle (2.20) correspond alors à l'estimateur du maximum de la loi *a posteriori* dans un cadre bayésien et gaussien. Les erreurs peuvent dépendre de la localisation spatio-temporelle. De ce fait, la covariance associée à une erreur-modèle mesure la dépendance du vecteur d'état évalué en deux points du domaine spatio-temporel. Nous reviendrons dans la section suivante sur la définition des matrices de covariances et sur leurs propriétés mathématiques.

Nous l'avons montré dans l'introduction de ce chapitre, le calcul de l'équation d'Euler-Lagrange associée à la fonctionnelle J de l'équation (2.20) mène à une impasse. Pour résoudre ce problème nous introduisons la variable auxiliaire suivante dans le calcul de la dérivée directionnelle de J :

$$\lambda(\mathbf{x}, t) = \int_A Q^{-1}(\mathbf{x}, t, \mathbf{x}', t') \left(\frac{\partial \mathbf{X}}{\partial t} + M(\mathbf{X})\right)(\mathbf{x}', t') \, d\mathbf{x}' dt' \tag{2.21}$$

ce qui conduit au système d'équation suivant :

$$\lambda(\mathbf{x}, t_1) = 0 \tag{2.22}$$

$$-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial M}{\partial \mathbf{X}}\right)^* \lambda = -\int_A \left(\frac{\partial H}{\partial \mathbf{X}}\right)^* R^{-1} H(\mathbf{X}, \mathbf{Y})(\mathbf{x}', t') \, d\mathbf{x}' dt' \tag{2.23}$$

$$\mathbf{X}(\mathbf{x}, t_0) = \int_\Omega B\lambda(\mathbf{x}', 0) \, d\mathbf{x}' + \mathbf{X}_0(\mathbf{x}) \tag{2.24}$$

$$\frac{\partial \mathbf{X}}{\partial t} + M(\mathbf{X}) = \int_A Q\lambda(\mathbf{x}', t') \, d\mathbf{x}' dt' \tag{2.25}$$

On consultera [**?**] (voir annexe A.4) qui détaille le calcul complet. Néanmoins, le calcul de la variable adjointe nécessite la connaissance du vecteur d'état (équations (2.22) et (2.21)) et vice versa (équations (2.24) et (2.25)). Pour résoudre ce problème, on utilise une décomposition incrémentale du vecteur d'état : $\mathbf{X} = \mathbf{X}_b + \delta\mathbf{X}$, le terme $\delta\mathbf{X}$ est l'incrément et $\mathbf{X}_b$ est la variable *background*. Les opérateurs M et H sont linéarisés par un développement limité d'ordre 1 et nous obtenons le système final :

$$\mathbf{X}_b(\mathbf{x}, t_0) = \mathbf{X}_b \tag{2.26}$$

$$\frac{\partial \mathbf{X}}{\partial t} + M(\mathbf{X}_b) = 0 \tag{2.27}$$

$$\lambda(\mathbf{x}, t_1) = 0 \tag{2.28}$$

$$-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial M}{\partial \mathbf{X}_b}\right)^* \lambda = -\int_A \left(\frac{\partial H}{\partial \mathbf{X}_b}\right)^* R^{-1} \left(H(\mathbf{X}_b, \mathbf{Y}) + \frac{\partial H}{\partial \mathbf{X}}(\delta\mathbf{X})\right) d\mathbf{x}' dt' \tag{2.29}$$

$$\delta\mathbf{X}(\mathbf{x}, t_0) = \int_\Omega B\lambda(\mathbf{x}', 0) \, d\mathbf{x}' \tag{2.30}$$

$$\frac{\partial \delta\mathbf{X}}{\partial t} + \frac{\delta M}{\mathbf{X}_b}(\delta\mathbf{X}) = \int_A Q\lambda(\mathbf{x}', t') \, d\mathbf{x}' dt' \tag{2.31}$$

Cet algorithme fonctionne ainsi : on calcule une première valeur pour le vecteur d'état $\mathbf{X}_b$ grâce aux équations (2.26) et (2.27) ; le terme incrémental est initialement supposé nul ce qui nous permet de calculer la variable adjointe par l'intermédiaire des équations (2.28) et (2.29) ; le vecteur incrémental est ensuite calculé avec les équations (2.30) et (2.31) ; le vecteur d'état est mis à jour avec le vecteur incrémental ; le procédé peut alors être itéré (calcul de la variable adjointe, puis calcul du vecteur incrémental) jusqu'à convergence de la méthode. On trouvera dans [**?**] un algorithme complet en page 8. On peut voir cet algorithme comme une formulation "faible" du 4DVar classique. Nous le baptiserons 4DVarMI dans la suite du manuscrit, MI signifiant Modèle Imparfait.

2.2 Matrices de covariance

Les matrices de covariance jouent un rôle très important dans le formalisme de l'assimilation de données. Dans le cadre fonctionnel, le vecteur d'état Z est une fonction $\mathbf{x} \mapsto Z(\mathbf{x})$ et on défini sa matrice de covariance par $\Sigma(\mathbf{x},\mathbf{x}') = \int (Z(\mathbf{x}) - \mathbb{E}Z(\mathbf{x}))^T (Z(\mathbf{x}') - \mathbb{E}Z(\mathbf{x}')) \mathrm{d}P_{Z(\mathbf{x}),Z(\mathbf{x}')}$ où $\mathbb{E}$ est l'espérance et $\mathrm{d}P_{Z(\mathbf{x}),Z(\mathbf{x}')}$ la mesure associée à la loi jointe de $(Z(\mathbf{x}), Z(\mathbf{x}'))$. Le vecteur Z est donc vu comme un vecteur aléatoire paramétré par $\mathbf{x}$ et Σ est donc une matrice d'autocovariance. L'inverse d'une matrice de covariance Σ est défini ainsi :

$$\int \Sigma^{-1}(\mathbf{x},\mathbf{x}'')\Sigma(\mathbf{x}'',\mathbf{x}')\mathrm{d}\mathbf{x}'' = \delta(\mathbf{x}-\mathbf{x}') \tag{2.32}$$

où δ est la fonction de Dirac. Le calcul de l'inverse d'une matrice est donc difficile mais nous pouvons l'établir dans des cas simples ou particuliers. Par exemple, si nous considérons une covariance de Dirac, c'est-à-dire $\Sigma(\mathbf{x},\mathbf{x}') = \delta(\mathbf{x},\mathbf{x}')$, l'inverse est également une Dirac. Considérons la fonctionnelle $\iint \mathrm{F}(Z(\mathbf{x}))^T \Sigma^{-1}(\mathbf{x},\mathbf{x}')\mathrm{F}(Z(\mathbf{x}'))\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{x}'$, pour une covariance de Dirac, cette fonctionnelle se simplifie en $\iint \|\mathrm{F}(Z(\mathbf{x}))\|^2 \mathrm{d}\mathbf{x}$. En terme de minimisation de fonctionnelle, choisir une covariance de Dirac est équivalent à une régularisation d'ordre 0. Sinon on considère une covariance exponentielle, c'est-à-dire $\Sigma(\mathbf{x},\mathbf{x}') = \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|}{\sigma})$, nous obtenons alors $\iint \mathrm{F}(Z(\mathbf{x}))^T \Sigma^{-1}(\mathbf{x},\mathbf{x}')\mathrm{F}(Z(\mathbf{x}'))\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{x}' = \frac{1}{2\sigma}\int (\|\mathrm{F}(z)\|^2 + \sigma^2 \left\|\nabla \mathrm{F}(Z)\right\|^2) \mathrm{d}\mathbf{x}$. Une covariance exponentielle se traduit donc par une régularisation d'ordre 1 en terme de minimisation fonctionnelle. Enfin, si on considère une covariance gaussienne, $\Sigma(\mathbf{x},\mathbf{x}') = \exp(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{\sigma^2})$, on obtient $\iint \mathrm{F}(Z(\mathbf{x}))^T \Sigma^{-1}(\mathbf{x},\mathbf{x}')\mathrm{F}(Z(\mathbf{x}'))\mathrm{d}\mathbf{x}\mathrm{d}\mathbf{x}' = \frac{1}{\sigma\sqrt{\pi}}\int \sum_{n\geq 0} \frac{1}{n!}(\frac{\sigma}{2})^{2n} \left\| \frac{\partial^n \mathrm{F}(Z)}{\partial \mathbf{x}^n} \right\|^2 \mathrm{d}\mathbf{x}$. Une covariance gaussienne correspond alors à une régularisation à tout ordre. On consultera [**?**] pour le détail de ces calculs. Ces cas particuliers nous permettrons de faire des choix pour les matrices Q, B et R lors de la mise en œuvre de 4DVarMI.

2.3 Application au calcul du flot optique

L'algorithme 4DVarMI a été utilisé pour le calcul du flot optique dans des séquences vidéo [**?**, **?**]. Le vecteur d'état est donc un vecteur vitesse. Les observations sont constituées des images

de la vidéo à certains instants. Puis nous calculons pour tout instant le champ de vitesse qui respecte au mieux l'équation d'observation et l'équation d'évolution. Certaines difficultés ont du être levées.

La première d'entre elles est le choix des modèles M et H. Comme nous souhaitons calculer des vitesses, le vecteur d'état est la distribution des vecteurs vitesse dans l'image et nous observons une séquence d'images. Le modèle H, liant état et observation, est donc tout naturellement le transport de la luminosité dans la séquence d'images ou la contrainte du flot optique. Le modèle M lui décrit la dynamique de l'image. Nous allons considérer une vitesse constante le long de chaque trajectoire. Cela correspond à une certaine régularité temporelle du champ des vitesses d'un plan à l'autre de la séquence d'images. Le problème se formalise donc ainsi :

$$\begin{aligned}
\mathbf{w}(\mathbf{x}; t = t_0) &= \mathbf{w}_0(\mathbf{x}) + \mathcal{E}_B && (2.33)\\
\mathbf{w}_t + \nabla\mathbf{w}^T\mathbf{w} &= \mathcal{E}_Q && (2.34)\\
I_t + \nabla I^T\mathbf{w} &= \mathcal{E}_R && (2.35)
\end{aligned}$$

où $\mathbf{w} = (u \quad v)^T$ est le vecteur d'état, c'est-à-dire la vitesse et I le vecteur d'observation, c'est-à-dire l'image. La mise en œuvre du 4DVarMI conduit à implémenter le système suivant :

$$\begin{aligned}
\mathbf{w}(\mathbf{x}; t = t_0) &= \mathbf{w}_0(\mathbf{x}) && (2.36)\\
\mathbf{w}_t + \nabla\mathbf{w}^T\mathbf{w} &= 0 && (2.37)\\
\lambda(\mathbf{x}, t_1) &= 0 && (2.38)\\
-\lambda_t - \nabla\lambda^T\mathbf{w} - (\nabla^\perp\mathbf{w})^T\lambda &= -\nabla I R^{-1} \star (I_t + \nabla I^T(\mathbf{w} + \delta\mathbf{w})) && (2.39)\\
\delta\mathbf{w}(\mathbf{x}, t_0) &= B \star \lambda(\mathbf{x}, t_0) && (2.40)\\
\delta\mathbf{w}_t + \nabla\delta\mathbf{w}^T\mathbf{w} + \nabla\mathbf{w}^T\delta\mathbf{w} &= Q \star \lambda && (2.41)
\end{aligned}$$

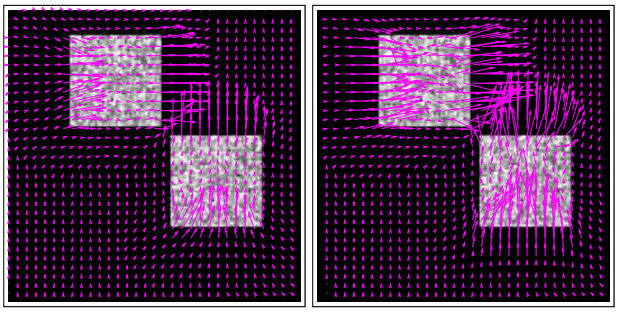
avec $\nabla^\perp\mathbf{w} = \begin{pmatrix} v_y & -u_y \\ -v_x & u_x \end{pmatrix}$. Pour obtenir ce système à partir du système d'équations (2.26–2.31) nous avons déterminé les adjoints des opérateurs $\frac{\partial M}{\partial \mathbf{w}}$ et $\frac{\partial H}{\partial \mathbf{w}}$ et considéré des covariances isotropiques, c'est à dire $\Sigma(\mathbf{x}, \mathbf{x}') = \Sigma(\mathbf{x} - \mathbf{x}')$, pour lesquelles l'équation (2.32) s'écrit maintenant comme une convolution. Remarquons que les trois cas particuliers examinés dans la section 2.2 sont isotropiques.

La seconde difficulté concerne la discrétisation et l'implémentation du système (2.36–2.41). En premier lieu, l'équation (2.37) est une équation 2D comportant un terme d'advection linéaire et un terme d'advection non linéaire. Obtenir un schéma numérique stable d'une telle équation est délicat. Nous avons proposé un schéma stable utilisant la technique de *spliting* et des schémas appropriés pour chaque type d'advection [**?**]. Parce qu'elle sont 2D, les équations (2.39) et (2.41) doivent également être découplées par *spliting*.

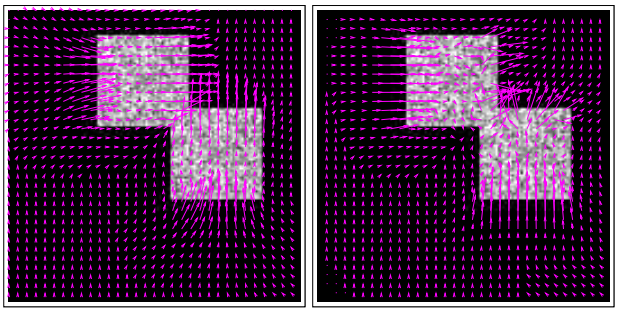
Une troisième difficulté concerne la stabilité du schéma incrémental. Les expériences montrent que le schéma incrémental diverge au bout d'un nombre, variable selon les expériences, d'itérations. La raison semble être que nous calculons de façon formelle les différentielles et adjoints des modèles d'évolution et d'observation, puis, les équations résultantes sont discrétisées. Or, il est établi que l'adjoint discrétisé d'un opérateur continu ne coïncide pas avec l'adjoint

de l'opérateur discrétisé. Notamment, il faut bien s'assurer que l'équation (2.17) soit vérifiée par les opérateurs discrets. Pour résoudre ce problème, nous calculons les différentielles et adjoints des modèles discrets à l'aide d'un logiciel de différentiation automatique, en l'occurence il s'agit ici de Tapenade[2] développée par l'équipe INRIA TROPICS. Cette approche permet non seulement d'obtenir la convergence de l'algorithme incrémental mais aussi de modifier plus facilement les modèles M et H sans avoir à procéder à des calculs complexes.

L'utilisation de la dynamique permet, comme nous le voyons dans les équations (2.36–2.41), de se passer de la régularisation du vecteur d'état (une telle régularisation apparaîtrait sous la forme d'un terme de diffusion sur la variable adjointe et le vecteur incrémental). La figure 2.1 montre un résultat typique. Dans cette séquence synthétique, deux carrés se déplacent avec un mouvement uniforme jusqu'à se rencontrer. Nous montrons un plan (c'est-à-dire une image à un instant donné) au début de la séquence et un second à la fin, lors de l'occultation. L'assimilation de données obtient un résultat moins lisse au contraire d'une méthode standard par régularisation. Néanmoins, il reste parfaitement possible de formaliser des contraintes de régularisation. En effet, la forme de l'équation (2.4) d'observation est très générale : il est possible d'y ajouter un terme portant sur le gradient du vecteur d'état qui, en conjonction d'une covariance de dirac par exemple, aboutisse à une régularisation du vecteur d'état comme nous l'expliquons dans [**?**] (voir annexe A.4 de ce document).



(a) Calcul au plan 4. Droite : 4DVarMI, gauche : [**?**].



(b) Calcul au plan 10. Droite : 4DVarMI, gauche : [**?**].

Figure 2.1 – Séquence synthétique : carrés s'occultant.

[2] http ://tapenade.inria.fr

L'équation d'évolution permet de calculer le vecteur d'état à tout instant à partir d'un vecteur initial et l'équation d'observation permet alors d'assimiler les observations dans le calcul du vecteur d'état. En poussant cette logique jusqu'au bout, nous pouvons remarquer qu'il n'est pas nécessaire d'utiliser les observations pour calculer $\mathbf{w}$ même si sans observations le résultat ne sera pas pertinent. Mais si une partie des observations est manquante, cela ne gênera aucunement le calcul du vecteur d'état. Ceci peut être facilement obtenu en définissant intelligemment la matrice R^{-1} : les observations (membre de droite de l'équation (2.39)) ne seront pas utilisées aux endroits où R^{-1} est proche de la matrice nulle. Nous gérons ainsi très naturellement le problème des données manquantes. Dans la figure 2.2, nous avons masqué une partie des observations pour simuler une donnée manquante. L'assimilation de données permet de donner un résultat cohérent fourni par l'équation d'évolution bien que celle-ci décrive très imparfaitement la dynamique de l'image.

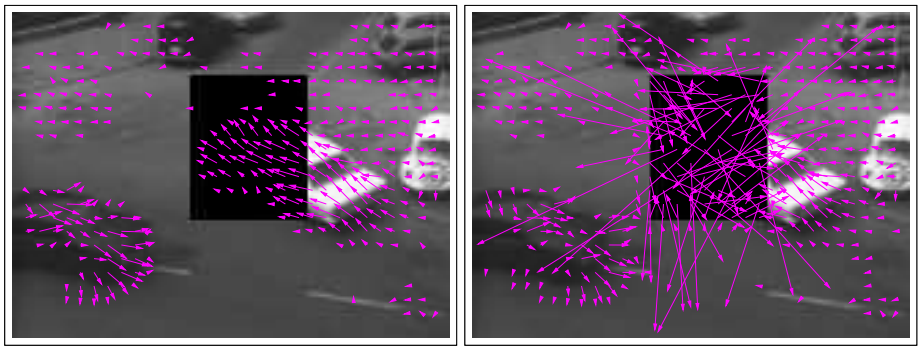
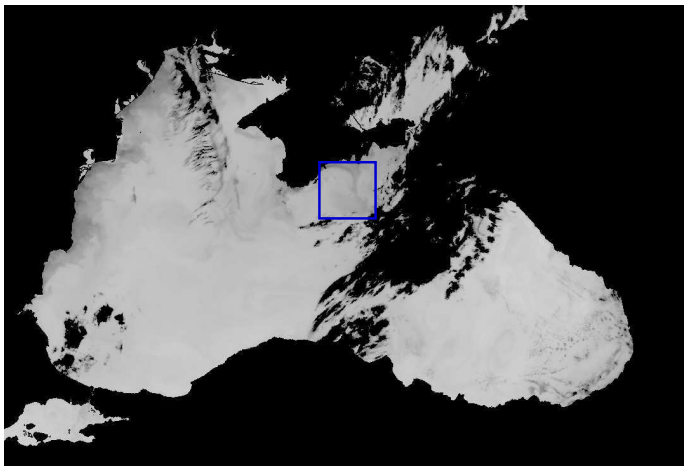


Figure 2.2 – Taxis de Hambourg et données manquantes. Droite : 4DVarMI, gauche : [**?**].

Nous avons testé également cet algorithme sur des données satellites sur lesquelles nous avons de véritables données manquantes : il s'agit de défauts d'acquisition ou de nuages occultant la scène [**?**]. Pour ces expériences, nous avons modifié le modèle d'observation H en utilisant non plus la contrainte de flot optique mais le transport de la luminosité, c'est-à-dire que $H(\mathbf{w}, I) = I(\mathbf{x}+\mathbf{w}, t+1) - I(\mathbf{x}, t)$ ce qui permet d'estimer les vecteurs vitesse ayant une norme élevée (voir aussi la sous-section 3.1). La figure 2.3 montre les données sur lesquelles nous avons testé l'algorithme : ce sont des images de température de surface. Le calcul a été fait dans la zone d'intérêt repérée en bleu. La figure 2.3 montre la situation générale au sixième plan d'acquisition sachant que la zone de calcul est occultée par un nuage dans le plan suivant (figure 2.3). Un algorithme standard ne pourra pas calculer le flot optique de façon pertinente là où les données sont manquantes. En revanche, notre approche le permet, comme le montre la figure 2.4, puisque les zones de données absentes sont identifiées et ignorées dans le calcul de la vitesse, cette dernière étant prise en charge par le modèle d'évolution.

2.4 Conclusion du chapitre

L'assimilation des données permet de résoudre une équation d'évolution sur l'état en tenant compte des observations directes ou indirectes du vecteur d'état. Pour employer ce formalisme



(a) Image complète et zone de calcul au plan 6.

(b) Zone de calcul aux plans 5, 6 et 7.

Figure 2.3 – Images SST et données manquantes.

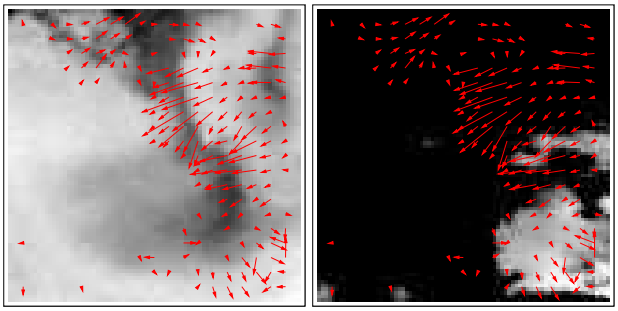


Figure 2.4 – Calcul du flot optique par 4DVarMI aux plans 5 et 6.

dans un cadre généraliste du traitement d'images, nous avons étendu l'équation d'observation dans le 4DVar. La dynamique des structures images étant, toujours dans un cadre général, inconnue, nous l'avons approchée avec un modèle exprimant la régularité temporelle et l'emploi d'un 4DVar autorisant une erreur sur l'équation d'évolution. À travers l'exemple du flot optique, nous avons essayé de montrer dans cette section que ces techniques peuvent apporter beaucoup au traitement d'images : elles permettent d'incorporer l'information temporelle dans les modèles et de gérer les données manquantes. Ces travaux ont d'abord été menés dans le cadre de ma délégation INRIA dans l'équipe-projet CLIME, Isabelle Herlin m'ayant demandé de réfléchir aux applications potentielles en traitement d'images. Depuis, nous poursuivons notre collaboration sur ce thème.
Bien que des résultats significatifs aient également été obtenus dans l'équipe-projet FLUMINANCE de l'INRIA sur le suivi d'objets et l'estimation du mouvement dans les images météorologiques, ce thème de recherche reste pour l'heure encore émergent dans le domaine du traitement des images.

L'assimilation des images satellitaires est un enjeu capital en sciences de l'environnement puisque seuls les satellites sont capables d'observer le globe terrestre dans sa totalité. Mais cette information, très riche, ne peut être directement assimilée dans les modèles de prévision car les grandeurs physiques mesurées dans les images ne sont pas les variables d'état des modèles. Des travaux sont actuellement menés, dans les équipes-projet CLIME et MOISE de l'INRIA pour ne citer qu'eux, pour bâtir des modèles intermédiaires qui ont pour rôle, toujours avec l'aide de l'assimilation de données, d'inférer certaines variables d'état à partir des images satellitaires. Il ne s'agit pas seulement de transformer l'information radiométrique issue du capteur en une variable d'état mais aussi d'exploiter l'information structurelle visible dans les images et là, le traitement d'images aura un grand rôle à jouer dans un futur proche.

Chapitre 3

Autres travaux

3.1 Méthodes numériques pour des problèmes non linéaires

Le calcul du flot optique est originellement un problème non linéaire puisqu'il s'agit de calculer le vecteur vitesse ou déplacement dans une équation de transport. Dans le papier originel de détermination du flot optique [**?**], les auteurs prennent cette équation de transport et la linéarisent par un développement de Taylor à l'ordre 1. Elle prend alors le nom de contrainte du flot optique. Toutefois, à cause de cette linéarisation, la détermination du champ des vitesses n'est possible que pour des petites vitesses. Dans ce chapitre nous étudions brièvement les méthodes numériques pour calculer un champ de vitesses à norme élevée et proposons deux contributions originales.

Méthodes incrémentales

Flot optique dual

Dans [**?**] est établi un schéma baptisé "Dual Optical Flow Scheme". Soit $\mathbf{w}$ la vecteur vitesse et $\tilde{\mathbf{w}}$ une estimation de ce dernier. Le vecteur $\mathbf{w}$ vérifie l'équation de contrainte du flot optique sur une séquence d'images $I(\mathbf{x},t)$:

$$\nabla I^T \mathbf{w} + I_t = 0 \tag{3.1}$$

Rappelons deux approximations :

$$\begin{aligned} I_t(\mathbf{x},t) &= I(\mathbf{x},t+1) - I(\mathbf{x},t) && (3.2)\\ I(\mathbf{x}+\tilde{\mathbf{w}},t+1) &= I(\mathbf{x},t) + \nabla I^T(\mathbf{x},t)\tilde{\mathbf{w}} + I_t(\mathbf{x},t) \\ &= \nabla I^T(\mathbf{x},t)\tilde{\mathbf{w}} + I(\mathbf{x},t+1) && (3.3)\end{aligned}$$

L'équation (3.2) est introduite dans (3.1), on remarquera qu'en pratique, on calcule le gradient temporel de I de cette façon. Puis on fait apparaître le terme $I(\mathbf{x}+\tilde{\mathbf{w}},t+1)$ et on sub-

stitue (3.3) :

$$\begin{aligned} \nabla I^T \mathbf{w} &+ I(\mathbf{x}, t+1) - I(\mathbf{x}, t) = 0 \\ \nabla I^T \mathbf{w} &+ I(\mathbf{x}, t+1) - I(\mathbf{x}+\tilde{\mathbf{w}}, t+1) + I(\mathbf{x}+\tilde{\mathbf{w}}, t+1) - I(\mathbf{x}, t) = 0 \\ \nabla I^T (\mathbf{w}-\tilde{\mathbf{w}}) &+ I(\mathbf{x}+\tilde{\mathbf{w}}, t+1) - I(\mathbf{x}, t) = 0 \end{aligned}$$

Notons $\mathbf{I}_t(\tilde{\mathbf{w}}) = I(\mathbf{x}+\tilde{\mathbf{w}}, t+1) - I(\mathbf{x}, t)$, intuitivement il s'agit d'une dérivée temporelle calculée au temps t sur un plan déplacé par le vecteur $\mathbf{w}$. Nous obtenons alors une contrainte sur le flot *résiduel* $\mathbf{w}-\tilde{\mathbf{w}}$, baptisée équation duale du flot optique :

$$\nabla I^T (\mathbf{w}-\tilde{\mathbf{w}}) + \mathbf{I}_t(\tilde{\mathbf{w}}) = 0 \tag{3.4}$$

Comme le problème de l'estimation du flot résiduel est mal posé, on régularise en cherchant le minimum de la fonctionnelle de coût suivante :

$$J(\mathbf{w}) = \int d\mathbf{x} \left((\nabla I^T (\mathbf{w}-\tilde{\mathbf{w}}) + \mathbf{I}_t(\tilde{\mathbf{w}}))^2 + \alpha^2 \nabla \mathbf{w}^T \nabla \mathbf{w} \right) \tag{3.5}$$

Méthode incrémentale

On retrouve ce schéma sur le flot optique résiduel dans [**?**, **?**] formulé cette fois sous une forme incrémentale. Le champ de vecteurs vitesses $\mathbf{w}$ vérifie l'équation de transport de la luminosité :

$$I(\mathbf{x}+\mathbf{w}(\mathbf{x}, t), t+1) = I(\mathbf{x}, t) \tag{3.6}$$

Soit $\tilde{\mathbf{w}}$ un estimé de $\mathbf{w}$, le vecteur incrémental est défini par : $\delta\mathbf{w} = \mathbf{w} - \tilde{\mathbf{w}}$. On remplace $\mathbf{w}$ dans l'équation (3.6) et on effectue un développement de Taylor à l'ordre 1 de I au voisinage de $\mathbf{x}+\tilde{\mathbf{w}}$:

$$\begin{aligned} I(\mathbf{x}+\tilde{\mathbf{w}}(\mathbf{x}, t) + \delta\mathbf{w}(\mathbf{x}, t), t+1) &= I(\mathbf{x}, t) \\ I(\mathbf{x}+\tilde{\mathbf{w}}(\mathbf{x}, t), t+1) + \nabla I^T (\mathbf{x}+\delta\mathbf{w}(\mathbf{x}, t), t+1) \delta\mathbf{w}(\mathbf{x}, t) &= I(\mathbf{x}, t) \\ \nabla I_{\mathbf{w}}^T \delta\mathbf{w} + \triangle_{\mathbf{w}} I &= 0 \end{aligned} \tag{3.7}$$

en notant $\nabla I_{\mathbf{w}} = \nabla I(\mathbf{x}+\mathbf{w}, t+1)$ et $\triangle_{\mathbf{w}} I = I(\mathbf{x}+\mathbf{w}, t+1) - I(\mathbf{x}, t)$. L'équation (3.7) est une équation de flot optique sur le terme incrémental, la quantité $\triangle_{\mathbf{w}} I$ jouant le rôle de dérivée temporelle et on retrouve l'équation (3.4).

Approche proposée

Plutôt que de faire apparaître le terme incrémental dans l'équation du flot optique puis de bâtir une fonctionnelle de coût autour de ce terme, nous proposons de le faire apparaître directement dans la fonctionnelle en utilisant le lemme suivant :

Lemme 1 *Soit* $\mathbf{w} \mapsto J(\mathbf{w})$ *une fonctionnelle, soit* $\mathbf{w}_0$ *une fonction relativement proche d'un minimum de* E. *Alors on a :*

$$\underset{\mathbf{w}}{\operatorname{argmin}} J(\mathbf{w}) = \underset{\delta\mathbf{w}}{\operatorname{argmin}} J(\mathbf{w}_0 + \delta\mathbf{w}) \tag{3.8}$$

Donc, plutôt que de minimiser $J(\mathbf{w})$ par rapport à $\mathbf{w}$, nous minimisons $J(\mathbf{w}+\delta\mathbf{w})$ par rapport à $\delta\mathbf{w}$. Cette nouvelle fonctionnelle s'écrit donc :

$$J(\mathbf{w}+\delta\mathbf{w}) = \int (I(\mathbf{x}+\mathbf{w}+\delta\mathbf{w}, t+1) - I(\mathbf{x}+\mathbf{w}, t))^2 + \alpha^2 \|\nabla(\mathbf{w}+\delta\mathbf{w})\|^2)\, d\mathbf{x} \tag{3.9}$$

Le premier intégrant est linéarisé en $\delta\mathbf{w}$ et nous minimisons la fonctionnelle suivante :

$$J(\mathbf{w}+\delta\mathbf{w}) = \int (\nabla I(\mathbf{x}+\mathbf{w}, t)^T \delta\mathbf{w} + \mathbf{I}(\mathbf{x}, \mathbf{w}, t))^2 + \alpha^2 \|\nabla(\mathbf{w}+\delta\mathbf{w})\|^2)\, d\mathbf{x} \tag{3.10}$$

avec $\mathbf{I}(\mathbf{x}, \mathbf{w}, t) = I(\mathbf{x}+\mathbf{w}, t+1) - I(\mathbf{x}, t) + I_t(\mathbf{x}+\mathbf{w}, t)$. Nous calculons la différentielle de J par rapport à $\delta\mathbf{w}$ et nous obtenons l'équation d'Euler-Lagrange suivante :

$$-\alpha^2\nabla^2\delta\mathbf{w} - \alpha^2\nabla^2\mathbf{w} + \nabla I(\mathbf{x}+\mathbf{w}, t)\,(\nabla I(\mathbf{x}+\mathbf{w}, t)^T \delta\mathbf{w} + \mathbf{I}(\mathbf{x}, \mathbf{w}, t)) = 0 \tag{3.11}$$

à résoudre par rapport à $\delta\mathbf{w}$. La fonction $\mathbf{w}$ est connue : 0 à la première itération, puis $\mathbf{w}+\delta\mathbf{w}$ avec les quantités calculées à l'itération précédente. L'équation (3.11) est linéaire en $\delta\mathbf{w}$, le laplacien est discrétisé comme dans [**?**] et on se ramène à un système linéaire à résoudre dont la matrice est à diagonale strictement dominante pour α strictement positif : le système est résolu en utilisant la méthode décrite dans [**?**], c'est-à-dire une méthode itérative de type Gauss-Seidel. Les méthodes à résolution directe sont inapplicables car le système est de trop grande taille : le nombre d'inconnues est le nombre de pixels de l'image soit plusieurs centaines de milliers.

Les méthodes incrémentales permettent donc, nous l'avons vu, de ramener un problème non linéaire en une suite de problèmes linéaires. Remarquons que c'est cette approche qui est utilisée en assimilation variationnelle de données pour le modèle imparfait. Dans la section suivante, nous proposons une seconde méthode, non incrémentale cette fois, pour résoudre le problème non linéaire de transport de la luminosité.

Méthodes directes

Il s'agit de résoudre directement l'équation de transport, équation (3.6), autrement dit de minimiser la fonctionnelle de coût de l'équation (3.9) (en rétablissant $\mathbf{w}$ à la place de $\mathbf{w}_0 + \delta\mathbf{w}$ dans cette fonctionnelle) sans avoir à linéariser le problème. Pour cela, il suffit de calculer directement la différentielle de E ce qui est parfaitement possible à condition de supposer que I soit différentiable. Ceci a déjà été fait, d'abord dans [**?**, **?**] pour les contours actifs puis dans [**?**] pour l'estimation du flot optique. Ces derniers utilisent à la fois une contrainte sur le transport de la luminosité et une contrainte sur le transport des contours. Pour notre part, nous considérons uniquement le transport de la luminosité, mais il est aisé de traiter le transport des contours en utilisant en données des images de contours ou de normes de gradients, et nous utilisons un schéma numérique alternatif.

Approche proposée

Calculons la différentielle de la fonctionnelle définie dans l'équation (3.9) par rapport à $\mathbf{w}$ et nous obtenons l'équation d'Euler-Lagrange suivante :

$$-\alpha^2\nabla^2\mathbf{w} + \nabla I(\mathbf{x}+\mathbf{w}, t+1)(I(\mathbf{x}+\mathbf{w}, t+1) - I(\mathbf{x}, t)) = 0 \tag{3.12}$$

Évidemment, cette équation n'est pas linéaire en $\mathbf{w}$ et nous ne pouvons pas utiliser les méthodes habituelles de résolution de système linéaire car le second membre dépend maintenant de $\mathbf{w}$. Plutôt que résoudre (3.12), nous résolvons :

$$\frac{\partial \mathbf{w}}{\partial t} - \alpha^2 \nabla^2 \mathbf{w} + \nabla I(\mathbf{x}+\mathbf{w}, t+1)(I(\mathbf{x}+\mathbf{w}, t+1) - I(\mathbf{x}, t)) = 0 \tag{3.13}$$

pour des fonctions $\mathbf{w} = \mathbf{w}(\mathbf{x}, t, \tau)$ et nous nous intéressons aux solutions stationnaires en τ qui sont également solutions de (3.12). Nous proposons d'utiliser le schéma numérique semi-implicite suivant :

$$\mathbf{w}^{k+1} = \mathbf{w}^k + \triangle\tau(\alpha^2 \nabla^2 \mathbf{w}^{k+1} - \nabla I(\mathbf{x}+\mathbf{w}^k, t+1)(I(\mathbf{x}+\mathbf{w}^k, t+1) - I(\mathbf{x}, t)))$$

en notant $\mathbf{w}^k(\mathbf{x}, t) = \mathbf{w}(\mathbf{x}, t, \tau = k)$ et $\triangle\tau$ le pas de discrétisation en τ. Le schéma est implicite sur le terme de diffusion ce qui mène à des schémas numériques inconditionnellement stables. Le schéma est explicite sur le terme de réaction, autrement il est impossible d'évaluer numériquement $I(\mathbf{x}, t+1)$ sur un vecteur $\mathbf{x}$ non connu. En regroupant les termes linéaires en $\mathbf{w}$, le schéma s'écrit :

$$(I - \triangle\tau\alpha^2\nabla^2)\,\mathbf{w}^{k+1} = \mathbf{w}^k - \triangle\tau\nabla I(\mathbf{x}+\mathbf{w}^k, t+1)(I(\mathbf{x}+\mathbf{w}^k, t+1) - I(\mathbf{x}, t)) \tag{3.14}$$

ce qui implique la résolution d'un système linéaire à chaque itération. Selon la valeur de $\triangle\tau\alpha^2$, nous obtenons une matrice à diagonale strictement dominante, ce qui nous autorise à utiliser une méthode de type Gauss-Seidel pour résoudre l'équation (3.14) à l'itération $k+1$.

Conclusion

Nous avons proposé deux contributions pour le calcul du flot optique dans le cas de vitesses élevées, cas pour lequel les méthodes basées sur la contrainte de flot optique deviennent trop approximatives. Ces méthodes peuvent facilement être adaptées à d'autres problèmes non linéaires. Ces travaux sont en cours de rédaction et seront prochainement soumis à publication.

3.2 Traitement d'images satellitaire : segmentation, dénombrement de cocotiers et occupations des sols dans les images IKONOS

Ces travaux concernent un problème d'occupation des sols dans le contexte particulier de la Polynésie française. La Polynésie Française est un archipel constitué d'un très grand nombre d'îles. Elles sont petites et très isolées. À cela on peut ajouter que la Polynésie Française, qui a le statut de Pays d'Outre-Mer (POM), est très dépendante de la Métropole. Il est important de développer les ressources propres de l'archipel. Le développement de ses ressources naturelles est l'une d'entre elles et parmi celles-ci, on trouve les cocoteraies. Les cocotiers poussent sur les îles et les atolls soit de façon sauvage, soit de façon artificielle au sein de plantations. On peut

également distinguer des plantations qui ont été laissées à l'abandon parce que le coût de l'entretien est élevé sur des îles désertes et éloignées de Papetee. Des cocotiers, on peut récupérer le bois pour le bâti. On peut aussi en extraire l'huile de coprah pour l'exportation et aussi récolter les noix de coco. Le service de l'Urbanisme de Papetee a souhaité faire le recensement des cocotiers sur l'ensemble de l'archipel. Nous avons été contacté par l'Université de la Polynésie Française, où est menée une activité de recherche sur la télédétection, en tant que spécialiste du traitement des images. L'utilisation d'images satellite est évidemment une réponse pertinente au problème du recensement des cocotiers car elle est effectuée à distance. Le coût humain (temps et accès aux îles) devient quasiment nul.

Le service de l'Urbanisme a fourni les données dont elle disposait : il s'agit d'images IKONOS, résolution d'un mètre au sol pour certaines et 80 centimètres pour d'autres, géoréférencées et acquises dans le canal visible. Mais elles ne possèdent pas de canal proche infrarouge (qui n'avait pas été acheté). Nous avons une dizaine d'images IKONOS sur différents Atolls, mais une image pèse très lourd. Par exemple, pour l'Atoll de Tikeheau, l'image fait 28517 pixels par 28617 pixels en RGB. Au final, nous disposons d'une grande quantité de données à traiter.

Dénombrement des cocotiers

C'est dans ce contexte que nous avons proposé un algorithme semi-supervisé pour dénombrer les cocotiers [**?**, **?**], la dernière référence est consultable à l'annexe A.5. La figure 3.1 résume brièvement la méthode. La première partie consiste à masquer les régions qui ne sont pas de la végétation. Habituellement, on utilise le canal proche infrarouge pour calculer le *Normalized Difference Vegetation Index* (NVDI), un indice de végétation pertinent. Mais nous ne disposons pas de l'infrarouge. Alternativement, nous avons proposé d'utiliser conjointement une classification bayésienne et une analyse de la texture pour discriminer les zones de végétation du reste de l'image. Cette étape est supervisée (apprentissage du classifieur bayésien et calibration des filtres de textures). La seconde partie est la segmentation des cocotiers sur les zones de végétation. L'image RGB est fusionnée en une image monochrome en utilisant la première composante ACP et les zones non végétatives sont masquées. La segmentation est basée sur une ligne de partage des eaux et dont les points d'immersion sont soigneusement choisis afin d'éviter une sur-segmentation. Ces points correspondent aux sommets des houppiers des cocotiers et sont caractérisés par un maximum local de la luminosité dans quatre directions. Ils sont localisables dans un voisinage 7×7. On identifie de façon similaire un réseau de minima locaux qui vont permettre d'individualiser les cocotiers. La figure 3.2 montre différentes étapes de l'algorithme appliquée sur une parcelle de l'Atoll de Tikeheau. Pour évaluer l'efficacité de la méthode une validation visuelle a été effectuée ainsi qu'une mission terrain sur l'Atoll de Tikeheau. Diverses parcelles ont été visitées sur lesquelles les cocotiers ont été dénombrés et leur agencement spatial noté. À partir des résultats de segmentation nous créons des cartes de densité comme celle montrée dans la figure 3.3 et elles sont confrontées à celles obtenues sur le terrain. Les résultats ont montré que la méthode est robuste. Elle a été appliquée à l'ensemble des données à notre disposition. La thèse de Raimana Teina [**?**] détaille les protocoles utilisés pour la validation et énumère dans ses annexes les résultats obtenus sur la dizaine d'Atoll traités. Les résultats ont été communiqués au Service de l'Urbanisme pour exploitation.

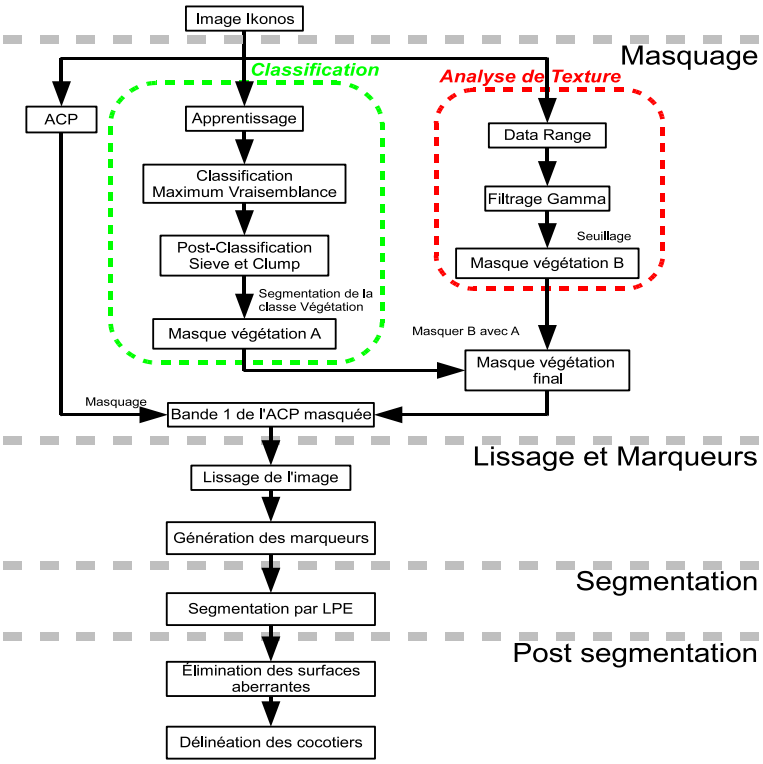
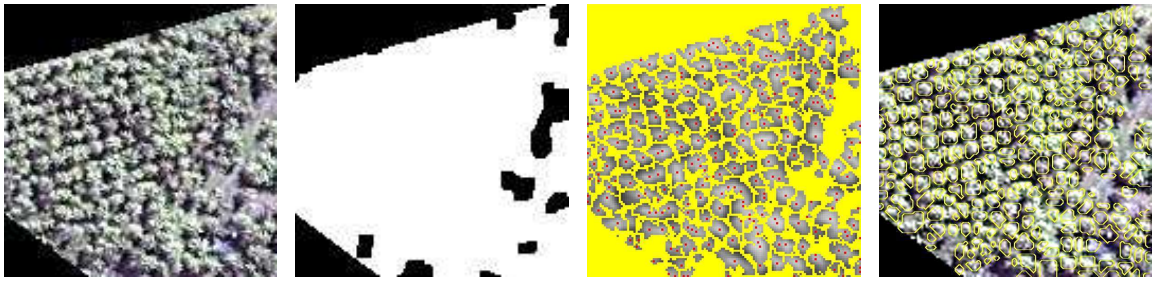


Figure 3.1 – Dénombrement des cocotiers.

Caractérisation des plantations

La seconde partie de la thèse a concerné la caractérisation des cocoteraies, c'est-à-dire dresser une typologie des configurations spatiales des plantations. Il s'agit de savoir quelles plantations sont entretenues, lesquelles sont sauvages et enfin lesquelles sont laissées à l'abandon. Pour cela nous avons proposé deux méthodes. La première méthode consiste à analyser directement la texture des images en utilisant des méthodes de classification [**?**], voir annexe A.5. Dans un premier temps, nous avons identifié l'information nécessaire à fournir aux classifieurs pour obtenir une classification correcte des secteurs tests en 3 classes de cocoteraies : sauvage, entretenue, abandonnée. Cette information est constituée des descripteurs de texture [**?**]. Nous avons également effectué une étude de performance sur différents classifieurs de l'état de l'art dont les bayésiens et les vastes marges. Ces méthodes donnent de bons résultats mais sont très lourdes car elles traitent pixel à pixel les données. Nous proposons une seconde méthode qui exploite directement la segmentation des cocotiers obtenue dans la section précédente.

Plutôt que d'analyser la texture des images, nous extrayons le centre des cocotiers segmentés puis nous analysons la répartition de ces points dans le plan. Nous avons proposé ainsi une méthode statistique [**?**], annexe A.5 du manuscrit, pour caractériser la répartition de ces points selon les trois classes de plantations précédemment citées. Un ensemble de cocotiers sera qualifié de sauvage si leurs centres suivent un processus ponctuel de Poisson c'est-à-dire que le nombre de points contenus dans une région B connexe quelconque suit une loi de Poisson de



(a) Composante 1 de l'ACP.

(b) Masque de végétation obtenu par classification et analyse de la texture.

(c) Maxima locaux (rouge) et réseau des minima locaux (jaune).

(d) Segmentation finale.

Figure 3.2 – Exemple de segmentation des cocotiers.

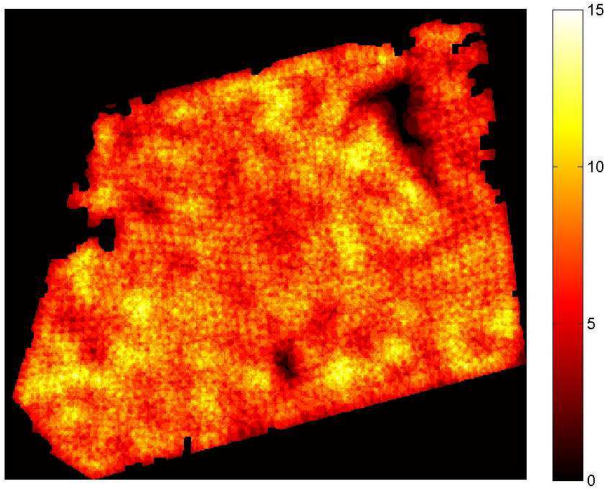
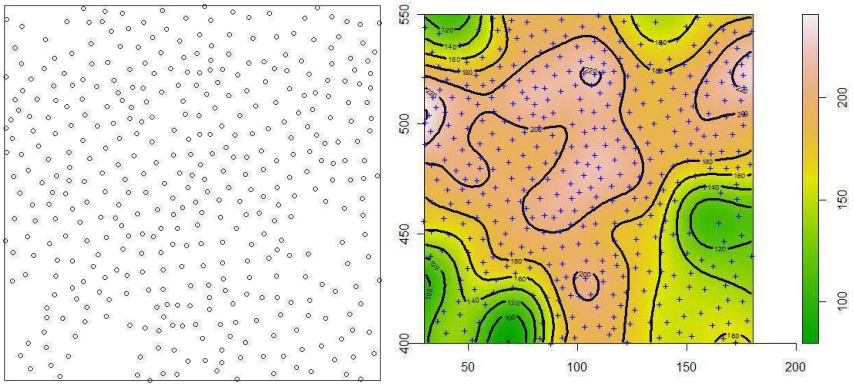


Figure 3.3 – Carte de densité.

paramètre $\lambda\mathcal{A}(B)$ où $\mathcal{A}(B)$ est l'aire de B et λ un paramètre réel caractérisant la densité de répartition de ces points. La méthode consiste donc à utiliser un test statistique caractérisant ou non un processus poissonien sur une fenêtre. Si le test est vrai alors l'estimation du paramètre λ donne une indication sur la densité de la cocoteraie sachant qu'un forte densité indique une cocoteraie sauvage. Le test se fait localement, il suffit de parcourir la fenêtre de test sur le domaine pour obtenir une description complète. La figure 3.4 montre un résultat typique obtenu sur une région test.



(a) Image des centres des cocotiers (b) Analyse des densités spatiales

Figure 3.4 – Partition en fonction de la densité estimée (paramètre λ).

Conclusion

Nous avons présenté nos contributions en occupation des sols : une méthode originale pour dénombrer des cocotiers sans utiliser le canal infrarouge, deux méthodes de typologie des cocoteraies la première par classification d'attributs extraits de l'image et la seconde utilisant une approche statistique caractérisant les répartitions spatiales des centres des cocotiers. Ces travaux ont été réalisés dans le contexte de l'encadrement de la thèse de Raimana Teina [**?**], dont j'ai assuré la plus grande part de la direction scientifique. La première contribution a donné lieu à une exploitation de toutes les images disponibles. Les résultats ont été communiquées par la suite à un service d'exploitation de l'administration polynésienne, ce qui, dans le contexte d'une thèse, est remarquable.

3.3 Suivi d'objets, approche probabiliste

Cet axe de recherche concerne le problème du suivi d'objets multiples. Le cas d'étude est le suivant : nous disposons, à différents instants, de mesures sur des objets en mouvement. Le nombre de ces mesures peut varier et cela pour deux raisons : 1/ le nombre de cibles n'est pas connu, il évolue au cours du temps, généralement à cause d'occultations ou de collisions entre ces cibles, 2/ les mesures contiennent des erreurs : certaines sont des fausses alarmes, d'autres

sont défaillantes. Nous supposerons également que les mesures sont des coordonnées d'objets. Ces mesures peuvent venir d'images mais pas nécessairement. Ce type de problème appartient à la classe des problèmes cités dans l'introduction de ce manuscrit : il est mal posé car le suivi des cibles nécessite au préalable l'association des mesures à ces cibles. Il n'y a pas de solution unique si on ne peut pas distinguer les cibles entre elles.

Jusqu'à présent nous avons examiné les problèmes du point de vue de l'analyse fonctionnelle. Ce n'est évidemment pas l'unique cadre de résolution possible et, par exemple, les approches stochastiques sont largement employées. Dans cette section, le *filtrage particulaire* [**?**] a été utilisé pour mettre en œuvre le suivi des cibles. Comme l'assimilation variationnelle de données, le filtrage particulaire permet de résoudre les trois équations (2.1,2.2,2.3) mais exprimées de façon discrète en temps. Le modèle d'observation permet d'attribuer les mesures aux cibles. Pour prédire l'évolution des cibles, l'équation d'évolution est utilisée pour calculer la loi du vecteur d'état à l'instant souhaité. Le modèle d'évolution M peut être complexe et non linéaire et le calcul de la loi conditionnelle du vecteur d'état difficile, voire impossible. Le filtre particulaire consiste à échantillonner le vecteur d'état (aux instants connus) formant ainsi un nuage de *particules*. On applique alors la loi des grands nombres pour en déduire une approximation de la loi conditionnelle du vecteur d'état. Remarquons alors deux différences importantes entre le filtrage particulaire et les algorithmes 4DVar : d'une part la formulation est discrète en temps pour la première contrairement à la seconde, d'autre part 4DVar calcule une solution optimale sur l'intervalle de temps des observations alors qu'en filtrage particulaire on calcule l'état au fur et à mesure que le temps s'écoule (comme un filtre de Kalman).

La première contribution à ces travaux est un algorithme permettant la mise en correspondance des mesures (avec données manquantes) avec les cibles. Le vecteur d'état est l'ensemble des trajectoires de chaque cible. Plutôt que de discriminer les cibles entre elles à partir de leur aspect, nous le faisons par leur trajectoire. Pour une cible nous construisons une trajectoire probable constituée des positions passées et futurs (que l'on obtient par prédiction grâce au modèle d'évolution). Puis, à l'aide de trois critères géométriques, nous retenons la trajectoire des observations la plus proche de la trajectoire précédemment construite [**?**, **?**, **?**].

La seconde contribution est de mettre en œuvre le filtrage particulaire pour le suivi multicible. Pour cela, nous utilisons l'algorithme géométrique d'association précédemment décrit. Mais nous devons disposer aussi d'un modèle d'évolution. Nous nous plaçons toujours dans le cas où la dynamique du vecteur d'état n'est pas connue. À partir de conditions initiales de l'état, nous avons proposé d'utiliser des B-splines interpolant ces premiers états. La B-spline est extrapolée pour fournir les prédictions [**?**]. Ce type de modèle construit des trajectoires relativement régulières mais malgré tout autorise des modifications importantes de la dynamique.

Cet algorithme de suivi a été adapté pour traiter le cas d'objets déformables. L'objet déformable est représenté par son contour, lequel est décomposé en un ensemble de cœfficients de Fourier formant le vecteur d'état [**?**]. L'aspect déformable est géré par le fait que l'algorithme puisse faire apparaître ou disparaître des trajectoires au cours du temps. Une autre application est le suivi dans les séquences d'images multimodales. Nous utilisons l'information mutuelle pour mettre en correspondance des zones d'intérêts, issues de modalités d'acquisition différentes, mais qui contiennent une conformation spatiale similaire. Chaque zone d'image con-

stitue alors la cible à suivre [**?**].

Ces travaux ont été effectués dans le cadre de la thèse d'Abir El Abed [**?**] dont j'ai participé de façon plus marginale à l'encadrement.

Chapitre 4

Conclusion et perspectives

Dans ce manuscrit nous avons résumé les deux axes principaux de nos recherches depuis 2003. Ces recherches ont en commun l'estimation du mouvement en utilisant un formalisme variationnel de résolution et le but d'intégrer l'information *a priori*, qu'elle soit spatiale ou temporelle, dans les formalismes de résolution des problèmes de traitement de l'image.

Le premier de ces axes est l'imagerie biologique dans le contexte d'objets sphériques. De cette manière nous montrons comment utiliser l'information liée au contexte expérimental, ce que nous appelons l'*a priori*, pour résoudre un problème mal posé. En ce sens, nous répondons à la première question posée dans l'introduction pour un cas particulier. Cette réponse dépend évidemment du contexte et il n'est pas possible d'y répondre de façon générale. Nous avons présenté nos contributions sur le suivi d'objets sur surface sphérique, la visualisation de structures évoluant sur ces surfaces et enfin plusieurs techniques de reconstruction de séquence d'images tridimensionnelles à partir d'une séquence d'images bidimensionnelles et d'une image tridimensionnelle initiale. Certaines de ces techniques exploitent la sphéricité et d'autres non. Ces algorithmes ont une spécificité par rapport aux autres méthodes de l'état de l'art : elles fonctionnent en fusionnant les données 2D et 3D, la reconstruction 3D est déduite de l'image tridimensionnelle par compensation du mouvement et contrainte de façon à s'aligner au mieux sur des observations bidimensionnelles. C'est aussi leur point faible car elles nécessitent aussi une première image 3D. Nous avons expliqué que le procédé de reconstruction 3D n'était pas fonctionnel pour des raisons liées au contexte expérimental et nous avons dû illustrer les algorithmes de reconstruction 3D sur des données synthétiques. Ceci n'est évidemment pas satisfaisant si on souhaite évaluer le réel apport de ces méthodes à la communauté du traitement d'image. Pour cette raison, nous devons travailler sur le problème de la reconstruction d'une image tridimensionnelle à partir d'une série d'acquisitions bidimensionnelles réalisées à différentes profondeurs optiques et nous avons détaillé une heuristique qui, toujours en exploitant l'information *a priori*, doit permettre d'autocalibrer les paramètres de la déconvolution à partir d'une mire synthétique déduite de la scène expérimentale. Ces travaux ont été mis en parenthèses pendant 3 ans mais nous souhaitons maintenant y revenir par le biais d'un en-

cadrement doctoral.

Le second axe fort de ces recherches concerne l'assimilation variationnelle de données qui est émergent dans le domaine du traitement de l'image. L'assimilation de données, nous l'avons vu, permet d'incorporer dans les problèmes l'information temporelle sous la forme d'une équation d'évolution du vecteur d'état. Néanmoins, cela demande de connaître la dynamique du vecteur d'état. Concernant le calcul du flot optique, nous nous sommes placé dans le contexte où l'on ne dispose pas de connaissances précises sur la dynamique des structures images. Un modèle très approximatif tel que l'invariance du vecteur vitesse le long d'une trajectoire permet néanmoins de gérer les problèmes d'occultations ou de données manquantes. Ainsi, nous répondons d'une façon assez générale aux deux questions posées dans l'introduction sur l'exploitation de l'information temporelle. Nos contributions sont une généralisation du modèle d'observation dans l'algorithme du 4DVarMI et ceci pour mettre en œuvre des algorithmes d'estimation du mouvement, une étude des matrices de covariance et de leur action dans la fonctionnelle de coût, deux algorithmes d'estimation du flot optique qui gèrent l'absence partielle de données d'observation. Les algorithmes proposés ont nécessité de longs développements car plusieurs difficultés ont été rencontrées sur la stabilité des schémas numériques, mais nous disposons maintenant de codes stables et fonctionnels. Ceci nous permet d'envisager d'autres développements.

En premier lieu nous pouvons proposer d'autres applications : une première application évidente est d'utiliser l'assimilation de données pour procéder à la reconstruction des images 3D+t par à partir des séquences 2D comme décrit dans la première partie de ce manuscrit : le cadre s'y prête totalement puisque nous avons là un processus temporel et un modèle d'observation. Les contextes expérimentaux pour lesquels la dynamique est connue sont également des terrains d'applications idéaux. Pour revenir sur l'imagerie océanographique, nous pouvons utiliser un modèle *shallow-water* qui modélise l'évolution des courants et de la température sur la couche supérieure des mers.

En second lieu, nous devrons affiner le modèle d'évolution dans les situations où la dynamique n'est pas connue. Le modèle que nous avons proposé décrit bien les cas pour lesquels le vecteur d'état évolue de façon régulière mais n'est pas capable de prendre en compte les irrégularités ou discontinuités temporelles. Une réponse possible est la régularisation : en introduisant un terme régularisant sur le vecteur d'état dans la fonctionnelle d'énergie du 4DVar et en choisissant correctement la matrice de corrélation associée à l'équation d'évolution, on peut alors limiter son influence, voire la supprimer totalement, dans le calcul du vecteur d'état. Sans équation d'évolution, l'équation d'observation et la régularisation spatiale permet d'obtenir une solution aux instants où le modèle d'évolution ne décrit plus correctement la dynamique.

En troisième lieu, il faut réfléchir aux stratégies alternatives pour formaliser le problème à résoudre en terme d'assimilation de données et cela pour permettre d'assimiler les images satellites pour les modèles physiques. En effet, la difficulté majeure à résoudre est que les grandeurs directement observées dans ces images ne sont pas des variables d'état de ces modèles. Nous avons déjà commencé significativement ce processus de réflexion sur le problème du flot optique. Donnons un exemple pour fixer les idées : l'équation du flot optique est une advection de la luminosité par le vecteur vitesse, elle peut donc être considérée comme une équation

d'évolution sur la luminosité. Cela signifie que la luminosité peut être traitée comme une composante du vecteur d'état, les autres composantes étant le vecteur vitesse. Cela a deux conséquences : la première est que les observations sont les images et non plus leur gradient spatio-temporel et que l'opérateur d'observation est maintenant une simple discrépance ; la seconde est que nous sommes capables de reconstruire l'image à n'importe quel instant que ce soit à des instants intermédiaires ou à des instants où l'observation est manquante. Nous avons pu implémenter et vérifier l'intérêt d'un tel modèle ; un article traitant ce sujet va être prochainement soumit. Ce type d'approche est donc pertinent en météorologie ou océanographie puisque le modèle comporte maintenant une variable d'état qui peut être directement observée.

Enfin, un dernier exemple important d'application concerne la cancérologie : des progrès significatifs ont été accomplis pour décrire l'évolution de certains tissus tumoraux sous la forme d'équations différentielles. Là encore, l'assimilation de données sera très utile pour relier ces modèles d'évolutions aux observations médicales de ces maladies. Tout comme en imagerie satellitaire, il faudra proposer des algorithmes qui soient capables d'extraire les bonnes informations des images médicales pour nourrir ces modèles de croissance de tumeurs. En collaboration avec l'équipe CNRS IPAL, nous avons élaboré un sujet de thèse prospectif sur ce thème, pour le cancer du sein et pour les cellules souches neuronales, qui sera proposé début 2011.

Bibliographie

[Béréziat & Berroir, 2000] Béréziat, D. & Berroir, J.-P. (2000). Motion estimation on meteorological infrared data using a total brightness invariance hypothesis. *Environnement Modeling System*, 15(15), 513–519.

[Béréziat & Herlin, 2008] Béréziat, D. & Herlin, I. (2008). Calcul du flot optique par assimilation variationnelle de données. In *Reconnaissance des Formes et Intelligence Artificielle (RFIA)* (pp. 341–350). Amiens (France).

[Béréziat & Herlin, 2009] Béréziat, D. & Herlin, I. (2009). Solving ill-posed problems using data assimilation. Application to optical flow estimation. In *International Conference on Computer Vision Theory and Application (VISAAP)*, volume 2 (pp. 594–602). Lisboa (Portugal).

[Béréziat & Herlin, 2010a] Béréziat, D. & Herlin, I. (2010a). Solving ill-posed image processing problems using data assimilation. *Numerical Algorithms*, to appear.

[Béréziat & Herlin, 2010b] Béréziat, D. & Herlin, I. (2010b). Using model of dynamics for large displacement estimation on noisy acquisitions. Soumis à Transactions on Image Processing.

[Béréziat et al., 2000] Béréziat, D., Herlin, I., & Younes, L. (2000). A generalized optical flow constraint and its physical interpretation. In *Computer Vision and Pattern Recognition* (pp. 487–492). Hilton Head Island (USA) : IEEE.

[Bergen et al., 1992] Bergen, J., Anandan, P., Hanna, K., & Hingorani, R. (1992). Hierarchical model-based motion estimation. *European Conference on Computer Vision (ECCV)*, (pp. 237–252).

[Brox & Weickert, 2002] Brox, T. & Weickert, J. (2002). Nonlinear matrix diffusion for optic flow estimation. *Pattern Recognition, Lecture Notes in Computer Science.*, 2449, 446–453.

[Chan & Vese, 2001] Chan, T. F. & Vese, L. A. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, 10(7).

[Cohen, 1991] Cohen, L. (1991). On active contour models with balloons. *Comp. Vision, Graphics and Image Processing : Image Understanding*, 53(2), 211–218.

[Doucet et al., 2000] Doucet, A., Godsill, S., & Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering. *Statistics and Computing*, 10(3), 197–2008.

[Duda & Hart, 1972] Duda, R. & Hart, P. (1972). Use of the hough transformation to detect lines and curves in pictures. *Comm. ACM*, 15, 11–15.

[El Abed, 2008] El Abed, A. (2008). *Suivi multi-objets par filtrage particulaire dans un contexte de données incomplète et/ou manquante.* PhD thesis, Pierre et Marie Curie.

[El Abed et al., 2006] El Abed, A., Dubuisson, S., & Béréziat, D. (2006). Comparison of statistical and shape-based approaches for non-rigid motion tracking with missing data using a particle filter. In *Advanced Concepts for Intelligent Vision Systems (ACVIS)* (pp. 185–196). Antwerp (Belgium).

[El Abed et al., 2007a] El Abed, A., Dubuisson, S., & Béréziat, D. (2007a). Association spatio-temporelle avec données manquantes par minimisation d'énergie. In *colloque Groupe Recherche et d'Étude en Traitement du Signal et de l'Image (GRETSI)* (pp. 65–68). Troyes (France).

[El Abed et al., 2007b] El Abed, A., Dubuisson, S., & Béréziat, D. (2007b). Energetic particle filter for online multiple target tracking. In *International Conference on Image Processing* (pp. 493–496). San Antonio, (USA).

[El Abed et al., 2007c] El Abed, A., Dubuisson, S., & Béréziat, D. (2007c). Energy minimization approach for online data association with missing data. In *International Conference on Computer Vision Theory and Application (VISAAP)* (pp. 371–378). Barcelona (Spain).

[El Abed et al., 2007d] El Abed, A., Dubuisson, S., & Béréziat, D. (2007d). ENMIM : Energetic normalized mutual information model for online multiple object tracking with unlearned motions. In *Advanced Concepts for Intelligent Vision Systems (ACVIS)* (pp. 955–967). Delft (the Netherlands).

[El Abed et al., 2008] El Abed, S., Dubuisson, S., & Béréziat, D. (2008). Energy association filter for online data association with missing data. *Communications in Computer and Information Science.*

[Forsey & Bartels, 1988] Forsey, D. & Bartels, R. (1988). Hierarchical B-spline refinement. *Computer Graphics*, 22(4), 205–212.

[Hadamard, 1923] Hadamard, J. (1923). *Lecture on Cauchy's Problem in Linear Partial Differential Equations.* New Haven : Yale University Press.

[Haralick, 1979] Haralick, H. (1979). Statistical and structural approaches to textures. In *Proceedings of the IEEE*, volume 67 (pp. 786–804).

[Horn & Schunck, 1981] Horn, B. & Schunck (1981). Determining optical flow. *Artificial Intelligence*, 17, 185–203.

[Kass et al., 1988] Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes : Active contour models. *International Journal on Computer Vision*, 1, 321–322.

[Le Dimet & Talagrand, 1986] Le Dimet, F.-X. & Talagrand, O. (1986). Variational algorithms for analysis and assimilation of meteorological observations : theoretical aspects. *Tellus*, 38A, 97–110.

[Odobez & Bouthemy, 1995] Odobez, J.-M. & Bouthemy, P. (1995). Robust multiresolution estimation of parametric motion models. *International Journal of Visual Communication and Image Representation*, 6(4), 348–365.

[Proesmans et al., 1994] Proesmans, M., Van Gool, L., Pauwels, E., & Oosterlinck, A. (1994). Determination of optical flow and its discontinuities using non-linear diffusion. In *European Conference on Computer Vision (ECCV)*, volume 2 (pp. 295–304).

[Rekik, 2007] Rekik, W. (2007). *Fusion de données temporelles, ou 2D+t, et spatiale, ou 3D, pour la reconstruction de scènes 3D+t et traitement d'images sphériques. Application à la biologie cellulaire.* PhD thesis, Université Pierre et Marie Curie.

[Rekik et al., 2003] Rekik, W., Béréziat, D., & Dubuisson, S. (2003). Suivi de structures évoluant sur une surface tridimensionnelle. In *Journée d'étude sur les méthodes pour les signaux complexes en traitement d'image* (pp. 159–165). Inria-Rocquencourt (France).

[Rekik et al., 2005] Rekik, W., Béréziat, D., & Dubuisson, S. (2005). MAPVIS : A map-projection based tool for visualizing scalar and vectorial information lying on spheroidal surfaces. In *International Conference on Information Visualisation* (pp. 636–641). London (UK) : IEEE.

[Rekik et al., 2006a] Rekik, W., Béréziat, D., & Dubuisson, S. (2006a). Image processing in spheric-shaped data using a geographical transformation. application to bio-cellular imaging. In *International Symposium on Biomedical Imaging (ISBI)* (pp. 1256–1259). Virginia (USA) : IEEE.

[Rekik et al., 2006b] Rekik, W., Béréziat, D., & Dubuisson, S. (2006b). Optical flow computation and visualization in spherical context. application on 3D+t bio-cellular sequences. In *Engineering in Medicine and Biology Society (EMBS)* (pp. 1645–1648). New-York (USA) : IEEE.

[Rekik et al., 2007] Rekik, W., Béréziat, D., & Dubuisson, S. (2007). 3D+t reconstruction in the context of locally spheric shaped data observation. In *International Conference on Computer Analysis of Images and Patterns (CAIP)* (pp. 482–489). Vienna (Austria).

[Rekik et al., 2008] Rekik, W., Béréziat, D., & Dubuisson, S. (2008). *Fusion de données spatiales (3D) et temporelles (2D+t) pour la reconstruction de scènes 3D+t : deux approches.* Technical report, Laboratoire d'Informatique de Paris 6.

[Rekik et al., 2010] Rekik, W., Béréziat, D., & Dubuisson, S. (2010). *Recovering 3D+t scenes for 3D and 2D+t data by motion compensation using a prior surface model.* Technical report, Laboratoire d'Informatique de Paris 6.

[Rekik et al., 2004] Rekik, W., Béréziat, D., Dubuisson, S., & Puff, N. (2004). Tracking of non-rigid structures evolving on 3D surfaces by compensating perspective view deformations. In *IASTED International Conference on Signal and Image Processing, SIP 2004* Honolulu, (USA).

[Teina, 2009] Teina, R. (2009). *Caractérisation de la cocoteraie des Tuamotu à partir d'images satellites à très haute résolution spatiale*. PhD thesis, Université Pierre et Marie Curie. `http://tel.archives-ouvertes.fr/tel-00516952/en/`.

[Teina et al., 2007] Teina, R., Béréziat, D., & Stoll, B. (2007). Étude de la cocoteraie de Tikehau sur des images Ikonos. In *MajecSTIC* (pp. 265–269). Caen (France).

[Teina et al., 2008] Teina, R., Béréziat, D., Stoll, B., & Chabrier, S. (2008). Toward a global Tuamotu Archipelago trees sensing using high resolution optical data. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, volume II Boston (USA).

[Teina et al., 2009] Teina, R., Béréziat, D., Stoll, B., & Chabrier, S. (2009). A comparative study of several supervised classifiers for coconut tree field's type mapping on 80 cm rgb pansharpened Ikonos images. In *Image Processing : Machine Vision Applications II. Proceedings of SPIE*, volume 7251 (pp. 72510X (9 pages)). San Jose (CA).

[Tikhonov, 1963] Tikhonov, A. N. (1963). Regularization of incorrectly posed problems. *Sov. Math. Dokl.*, 4, 1624–1627.

Chapitre 5

Curriculum vitæ

BÉRÉZIAT Dominique
Université Pierre et Marie Curie
Laboratoire d'Informatique de Paris VI
4 place Jussieu
75005 Paris
Né le 8 septembre 1968.

5.1 Carrière

2001–
Maître de conférence à l'Université Pierre et Marie Curie (Paris 6).

2006–2008
Délégation INRIA dans l'équipe-projet CLIME, INRIA Paris-Rocquencourt (Le Chesnay).

2000-2001
Ingénieur expert dans le projet AIR, INRIA Rocquencourt (Le Chesnay).

1998–2000
Attaché temporaire d'enseignement et de recherche (ATER) à l'Institut d'Informatique d'Entreprise (IIE, école d'ingénieur dépendant du CNAM, Évry).

5.2 Diplômes

1995–1999
Thèse en mathématique appliquée à l'Université Paris-Sud Orsay (financement MESR) sous la direction de Laurent Younes et dont le sujet est "Estimation et suivi de structures déformables en mouvement. Application à la météorologie". Mention Très Honorable.

1992–1993
Diplôme d'études approfondies "Modélisation stochastique et statistique", mention bien. Université Paris-Sud Orsay.

1990–1992
Maîtrise de Probabilités et Statistiques, Maîtrise d'Analyse Numérique. Université Paris-Sud Orsay.

5.3 Activités d'enseignement

Enseignements à l'UPMC

Ces cours ont été donnés en Informatique sauf mention contraire.

- **Introduction à l'imagerie numérique.** Cours, TD, TME. Enseignement de L3.
- **Initiation à l'automatisation des tâches.** TD, TME. Enseignement de L2.
- **Initiation à l'informatique pour scientifique.** TD, TME. Enseignement de L1.
- **Initiation au langage C** TD, TME. Enseignement de L1.
- **Analyse multiéchelles et ondelettes.** Cours. Enseignement de M2.
- **Acquisition et traitement des images médicales.** Cours. Enseignement de M2.
- **Méthodologies des images médicales.** Cours. Enseignement de M2.
- **Environnement de développement.** TD, TME. Enseignement de L3.
- **Initiation au traitement d'images et à l'infographie.** Cours, TD, TME. Enseignement de L3.
- **Programmation scientifique en Fortran 90.** Cours, TD, TME. Enseignement de L2.
- **Programmation parallèle et répartie.** Cours, TD, TME. Enseignement de M1.
- **Algorithmique graphique et animation.** Cours, TD, TME. Enseignement de M1.
- **Programmation récursive (scheme).** TME. Enseignement de L1.
- **Algorithmique Numérique.** Cours, TD et TME en Magister MIAIF.
- **Traitement des images médicales.** Cours de M2.
- **Méthodes numériques et Informatique pour les géosciences.** TD et TME en Licence de géologie.
- **Initiation à la programmation en Pascal et bureautique.** TD et TME en DEUG Sciences de la matière, Physique et Chimie.

Enseignements à IST

L'IST était une école d'Ingénieur attachée à l'UPMC. Elle s'est transformée en l'École Polytech'Paris-UPMC.

- **Infographie et mise en œuvre avec OpenGL**. Cours, TD, TME.
- **Parallélisme**. Cours, TD, TME.

Enseignements à IIE/CNAM

L'IIE est une école d'ingénieur dépendant du CNAM et basée à Evry. Elle s'appelle maintenant l'école nationale supérieure d'informatique pour l'industrie et l'entreprise (ENSIIE).

- **Programmation et algorithmique.** TD et TME. Enseignement de première année.
- **Encadrement de projets.** Enseignement de première année.
- **Programmation objet en C++ et Java.** TME. Enseignement de seconde année.
- **Vision par ordinateur.** Enseignement de troisième année.
- **Encadrement de projets en robotique.** Enseignement de seconde année.

Enseignements à l'ISTM

L'ISTM est une école d'ingénieur basée anciennement à l'Université Léonard de Vinci (à la date de ces vacations) puis à l'ESIEE (Marne-La-Vallée). Elle forme des ingénieurs ayant la double compétence Management/Informatique.

- **Programmation structurée en C.** Cours, TD, TME.
- **Reconnaissance de forme dans les images.** Enseignement de troisième année.
- **Encadrement de projet de fin de première année.**

5.4 Activité de recherches

Encadrement de thèses

Raimana Teina	2005–2009	Caractérisation de la cocoteraie des Tuamotu à partir d'images satellites à très hautes résolution spatiale. Bourse de l'école doctorale "EDITE de Paris". Taux d'encadrement : 70%.
Abir El Abed	2004–2008	Fusion de données pour la reconstruction 3D temporelle. Bourse de l'école doctorale "EDITE de Paris". Taux d'encadrement : 15%.
Wafa Rekik	2003–2007	Fusion de données temporelles, ou 2D+t, et spatiale, ou 3D, pour la reconstruction de scènes 3D+t et traitement d'images sphériques. Application à la biologie cellulaire. Bourse de l'école doctorale "EDITE de Paris". Taux d'encadrement : 80 %.

Encadrement de stages Master/DEA

Imen Jdey	2009	Mise en œuvre de techniques de différentiation automatique pour l'estimation de flot optique par assimilation de données. Internship INRIA (4 mois).
Sahar Syassi	2009	Implantation Matlab d'un algorithme d'assimilation variationnelle de données et étude de l'influence des matrices de covariances associées aux équations d'observation et d'évolution. Internship INRIA (4 mois).

Carlos De Almedia	2008	3D Reconstruction using Data Assimilation. Internship INRIA (4 mois.)
Masnou Amiri	2006	Étude de la morphologie des neurones en développement (6 mois).
Olivier Mougin	2005	Implémentation d'algorithmes d'interpolation par B-splines (4 mois).
Frédéric Tchéou	2005	Caractérisation par analyse d'image des stades initiaux de la bio-minéralisation nacrée chez les huîtres perlières (6 mois).
Raimana Teina	2005	Segmentation d'images bruitées et faiblement contrastées selon une approche par Level Sets (6 mois).
Richard Moulen	2004	Reconstruction tridimensionnelle par déconvolution automatique en microscopie cellulaire (6 mois).
Wafa Rekik	2003	Recalage géométrique et analyse de vésicules cellulaires en imagerie focale (6 mois).
Issam El Hattab	2003	Reconstruction tridimensionnelle de coupes d'images vésiculaires. Traitement des structures tridimensionnelles (6 mois).
Kamel Haddadou	2002	Segmentation des anévrismes de l'aorte dans des images scanner X (6 mois).
Jacopo Grazzini	2000	Étude des trajectoires et de la déformation de structures nuageuses dans des séquences d'images satellites (6 mois).

Vie scientifique

- Participations à des programmes nationaux de recherche :
  - ANR (programme APP TECSAN) "MICO", (2011-2014), 10%[1].
  - ANR (programme SYSCOMM) "Geo-FLUIDS" (2010–2013), 10 %.
  - ANR (programme Masse de données) "ADDISA" (2006-2009), 10 %.
- Relecteur régulier pour *Transaction on Image Processing, Transaction on Geosciences and Remote Sensing* et *Pattern Recognition Letters* et occasionnellement pour *International Conference on Image Processing*.
- Organisateur de la journée "Imagerie Biologique et Médicale de l'UPMC" (novembre 2004).
- Participation à l'organisation du congrès international SCAN (Paris, 2002).
- Participation à l'organisation du congrès international ICAOS (Paris, 1996).

5.5 Activités administratives

Responsabilités pédagogiques actuelles

- Responsable de l'UE "Introduction à l'image numérique" (et anciennement l'UE "Initiation au traitement d'image et à l'infographie").
- Responsable de l'UE "Analyse multiéchelles et ondelettes".
- Membre du jury d'admission pour la spécialité IMA (Master d'Informatique de l'UPMC).

[1] Taux de participation en temps "recherche"

Mobilité des étudiants

– Responsable du Programme International de Master “Management de l’Imagerie Biomédicale” en partenariat avec National University of Singapore et Asian Institute of Technology.

Comités de sélection

2004–2007 Membre de la commission de spécialistes CNU 27 de l’UPMC.

2008–2011 Membre du comité d’experts CNU 27 de l’UPMC.

5.6 Publications

Article de journaux

[1] D. Béréziat and J.-P. Berroir. Motion estimation on meteorological infrared data using a total brightness invariance hypothesis. *Environnement Modeling System*, 15(15) :513–519, June 2000.

[2] D. Béréziat, J.-P. Berroir, S. Bouzidi, and H. Herlin, I. ans Yahia. Land use and wind estimation as inputs for air pollution modelling. *ERCIM NEWS*, (34), 1998. Special Issue on Environmental Modelling.

[3] D. Béréziat and I. Herlin. Solving ill-posed image processing problems using data assimilation. *Numerical Algorithms*, to appear, 2010.

[4] D. Béréziat and I. Herlin. Using model of dynamics for large displacement estimation on noisy acquisitions. October 2010. Soumis à Transactions on Image Processing.

[5] J.-P. Berroir and D. Béréziat. Cloud cover estimation for air quality models using meteorological sensors. *System Analysis Modeling Simulation, (SAMS)*, 37(3) :321–328, 2000. Special issue on air pollution modelling.

[6] A. El Abed, S. Dubuisson, and D. Béréziat. *Advanced Concepts for Intelligence Vision Systems*, volume 4179/2006 of *Lecture Notes in Computer Science*, chapter Comparison of Statistical and Shape-Based Approaches for non-rigid Motion Tracking with Missing Data Using a Particle Filter, pages 185–196. Springer, 2006.

[7] I. Herlin, I. Cohen, and D. Béréziat. Wind estimation by image processing for air pollution modeling. *System Analysis Modeling Simulation, (SAMS)*, 32 :57–66, 1998.

[8] H. Yahia, J.-P. Berroir, I. Cohen, D. Béréziat, S. Bouzidi, and I. Herlin. Traitement d’images de séquences de données océanographiques. *Océanis, Edition de l’Institut Océanographique*, 2000.

Chapitre de livre

[1] S. El Abed, S. Dubuisson, and D. Béréziat. Energy association filter for online data association with missing data. *Communications in Computer and Information Science*, 2008.

Conférences internationales

[1] D. Béréziat and J.-P. Berroir. Use of image processing and remote sensing techniques to improve input data quality an collection. In *International Conference on Air Pollution Modelling and Simulation (APMS)*, pages 563–572, Marne-La-Vallée (France), October 1998. Elsevier.

[2] D. Béréziat and I. Herlin. Object based optical flow estimation with an affine prior model. In *International Conference on Pattern Recognition*, volume 3, pages 1060–1063, Barcelona (Spain), September 2000.

[3] D. Béréziat and I. Herlin. Solving ill-posed problems using data assimilation. Application to optical flow estimation. In *International Conference on Computer Vision Theory and Application (VISAAP)*, volume 2, pages 594–602, Lisboa (Portugal), Febuary 2009.

[4] D. Béréziat, I. Herlin, and L. Younes. Motion detection in meteorological images sequences : Two methods and their comparison. In *Proceedings of SPIE'97*, volume 3217, pages 332–341, London (UK), September 1997.

[5] D. Béréziat, I. Herlin, and L. Younes. Motion estimation using a volume conservation hypothesis. In *International Conference on Accoustic, Speech and Signal Processing (ICASSP)*, Phoenix (USA), March 1999.

[6] D. Béréziat, I. Herlin, and L. Younes. A generalized optical flow constraint and its physical interpretation. In *Computer Vision and Pattern Recognition*, pages 487–492, Hilton Head Island (USA), June 2000. IEEE.

[7] D. Béréziat, I. L. Herlin, G. Giraudon, C. Nguyen, and C. Graffigne. Segmentation of echocardiographic images with markov random fields. In *European Conference on Computer Vision (ECCV)*, volume 2, pages 201–206, Stockholm (Sweeden), May 1994.

[8] A. El Abed, S. Dubuisson, and D. Béréziat. Comparison of statistical and shape-based approaches for non-rigid motion tracking with missing data using a particle filter. In *Advanced Concepts for Intelligent Vision Systems (ACVIS)*, pages 185–196, Antwerp (Belgium), September 2006.

[9] A. El Abed, S. Dubuisson, and D. Béréziat. Energetic particle filter for online multiple target tracking. In *International Conference on Image Processing*, pages 493–496, San Antonio, (USA), September 2007.

[10] A. El Abed, S. Dubuisson, and D. Béréziat. Energy minimization approach for online data association with missing data. In *International Conference on Computer Vision Theory and Application (VISAAP)*, pages 371–378, Barcelona (Spain), March 8-11 2007.

[11] A. El Abed, S. Dubuisson, and D. Béréziat. ENMIM : Energetic normalized mutual information model for online multiple object tracking with unlearned motions. In *Advanced Concepts for Intelligent Vision Systems (ACVIS)*, pages 955–967, Delft (the Netherlands), August 2007.

[12] J. Grazzini, D. Béréziat, and I. Herlin. Analysis of cloudy structures evolution on meteorological satellite acquisitions. In *International Conference on Image Processing*, Thessaloniki (Greece), 7-10 october 2001.

[13] E. Huot, I. Herlin, and D. Béréziat. Segmentation of temporal effects on phasimetric SAR images. In *International Conference on Pattern Recognition*, volume II, pages 1390–1392, Brisbane (Australia), August 1998.

[14] W. Rekik, D. Béréziat, and S. Dubuisson. MAPVIS : A map-projection based tool for visualizing scalar and vectorial information lying on spheroidal surfaces. In *International Conference on Information Visualisation*, pages 636–641, London (UK), July 2005. IEEE.

[15] W. Rekik, D. Béréziat, and S. Dubuisson. Image processing in spheric-shaped data using a geographical transformation. application to bio-cellular imaging. In *International Symposium on Biomedical Imaging (ISBI)*, pages 1256–1259, Virginia (USA), April 2006. IEEE.

[16] W. Rekik, D. Béréziat, and S. Dubuisson. Optical flow computation and visualization in spherical context. application on 3D+t bio-cellular sequences. In *Engineering in Medicine and Biology Society (EMBS)*, pages 1645–1648, New-York (USA), September 2006. IEEE.

[17] W. Rekik, D. Béréziat, and S. Dubuisson. 3D+t reconstruction in the context of locally spheric shaped data observation. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, pages 482–489, Vienna (Austria), August 2007.

[18] W. Rekik, D. Béréziat, S. Dubuisson, and N. Puff. Tracking of non-rigid structures evolving on 3D surfaces by compensating perspective view deformations. In *IASTED International Conference on Signal and Image Processing, SIP 2004*, Honolulu, (USA), August 23-25 2004.

[19] R. Teina, D. Béréziat, and B. Stoll. A spatial poisson process to classify coconut fields on Ikonos pansharpened images. In *Multispectral, Hyperspectral, and Ultraspectral Remote Sensing Technology, Techniques, and Applications II. Proceedings of SPIE*, volume 7149, page 71491E (10 pages), Noumea (New Caledonia), November 2008.

[20] R. Teina, D. Béréziat, B. Stoll, and S. Chabrier. Toward a global Tuamotu Archipelago trees sensing using high resolution optical data. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, volume II, Boston (USA), July 2008.

[21] R. Teina, D. Béréziat, B. Stoll, and S. Chabrier. A comparative study of several supervised classifiers for coconut tree field's type mapping on 80 cm rgb pansharpened Ikonos images. In *Image Processing : Machine Vision Applications II. Proceedings of SPIE*, volume 7251, page 72510X (9 pages), San Jose (CA), January 2009.

Conférences nationales

[1] D. Béréziat and I. Herlin. Calcul du flot optique par assimilation variationnelle de données. In *Reconnaissance des Formes et Intelligence Artificielle (RFIA)*, pages 341–350, Amiens (France), 23–25 janvier 2008.

[2] A. El Abed, S. Dubuisson, and D. Béréziat. Association spatio-temporelle avec données manquantes par minimisation d'énergie. In *colloque Groupe Recherche et d'Étude en Traitement du Signal et de l'Image (GRETSI)*, pages 65–68, Troyes (France), septembre 2007.

[3] J. Grazzini, D. Béréziat, and I. Herlin. Analyse de l'évolution des structures nuageuses dans des séquences d'images satellitaires météorologiques. In *ORASIS*, Cahors (France), 5-8 juin 2001.

[4] W. Rekik, D. Béréziat, and S. Dubuisson. Suivi de structures évoluant sur une surface tridimensionnelle. In *Journée d'étude sur les méthodes pour les signaux complexes en traitement d'image*, pages 159–165, Inria-Rocquencourt (France), 9-10 décembre 2003.

[5] R. Teina, D. Béréziat, and B. Stoll. Étude de la cocoteraie de Tikehau sur des images Ikonos. In *MajecSTIC*, pages 265–269, Caen (France), 2007.

Autres publications

[1] D. Béréziat and I. Herlin. Solving ill-posed image processing problems using data assimilation. Application to optical flow. Research Report 6477, INRIA, March 2008.

[2] D. Béréziat and I. Herlin. Solving ill-posed image processing problems using data assimilation. Research Report 6879, INRIA, March 2009.

[3] D. Béréziat, I. L. Herlin, G. Giraudon, C. Nguyen, and C. Graffigne. Comparison of different markov field model for segmenting echocardiographic images. Technical Report 2424, INRIA, 1994.

[4] Dominique Béréziat. *Détection et suivi de structures déformables en mouvement : application à la météorologie*. PhD thesis, Université Paris Sud, octobre 1999.

[5] W. Rekik, D. Béréziat, and S. Dubuisson. Fusion de données spatiales (3D) et temporelles (2D+t) pour la reconstruction de scènes 3D+t : deux approches. Technical report, Laboratoire d'Informatique de Paris 6, juin 2008.

[6] W. Rekik, D. Béréziat, and S. Dubuisson. Recovering 3D+t scenes for 3D and 2D+t data by motion compensation using a prior surface model. Technical report, Laboratoire d'Informatique de Paris 6, August 2010.

Annexe $\mathcal{A}$

Sélection de publications

Nous présentons les publications suivantes :

[1] D. Béréziat and J.-P. Berroir. Motion estimation on meteorological infrared data using a total brightness invariance hypothesis. *Environnement Modeling System*, 15(15) :513–519, June 2000.

[2] W. Rekik, D. Béréziat, and S. Dubuisson. Image processing in spheric-shaped data using a geographical transformation. application to bio-cellular imaging. In *International Symposium on Biomedical Imaging (ISBI)*, pages 1256–1259, Virginia (USA), April 2006. IEEE.

[3] W. Rekik, D. Béréziat, and S. Dubuisson. Optical flow computation and visualization in spherical context. application on 3D+t bio-cellular sequences. In *Engineering in Medicine and Biology Society (EMBS)*, pages 1645–1648, New-York (USA), September 2006. IEEE.

[4] W. Rekik, D. Béréziat, and S. Dubuisson. Recovering 3D+t scenes for 3D and 2D+t data by motion compensation using a prior surface model. Technical report, Laboratoire d'Informatique de Paris 6, August 2010.

[5] W. Rekik, D. Béréziat, and S. Dubuisson. Fusion de données spatiales (3D) et temporelles (2D+t) pour la reconstruction de scènes 3D+t : deux approches. Technical report, Laboratoire d'Informatique de Paris 6, juin 2008.

[6] W. Rekik, D. Béréziat, and S. Dubuisson. 3D+t reconstruction in the context of locally spheric shaped data observation. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, pages 482–489, Vienna (Austria), August 2007.

[7] D. Béréziat and I. Herlin. Solving ill-posed image processing problems using data assimilation. *Numerical Algorithms*, to appear, 2010.

[8] D. Béréziat and I. Herlin. Using model of dynamics for large displacement estimation on noisy acquisitions. October 2010. Soumis à Transactions on Image Processing.

[9] R. Teina, D. Béréziat, and B. Stoll. A spatial poisson process to classify coconut fields on Ikonos pansharpened images. In *Multispectral, Hyperspectral, and Ultraspectral Remote Sensing Technology, Techniques, and Applications II. Proceedings of SPIE*, volume 7149, page 71491E (10 pages), Noumea (New Caledonia), November 2008.

[10] R. Teina, D. Béréziat, B. Stoll, and S. Chabrier. Toward a global Tuamotu Archipelago trees sensing using high resolution optical data. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, volume II, Boston (USA), July 2008.

[11] R. Teina, D. Béréziat, B. Stoll, and S. Chabrier. A comparative study of several supervised classifiers for coconut tree field's type mapping on 80 cm rgb pansharpened Ikonos images. In *Image Processing : Machine Vision Applications II. Proceedings of SPIE*, volume 7251, page 72510X (9 pages), San Jose (CA), January 2009.

A.1 Estimation du mouvement par invariance de la luminosité totale

[1] D. Béréziat and J.-P. Berroir. Motion estimation on meteorological infrared data using a total brightness invariance hypothesis. *Environnement Modeling System*, 15(15) :513–519, June 2000.

ELSEVIER

Environmental Modelling & Software 15 (2000) 513–519

Environmental Modelling & Software

www.elsevier.com/locate/envsoft

Motion estimation on meteorological infrared data using a total brightness invariance hypothesis

Dominique Béréziat *, Jean-Paul Berroir

INRIA-AIR Project, BP 105, 78153 Le Chesnay Cedex, France

Abstract

This work investigates a novel approach for cloud motion wind (CMW) estimation of Meteosat infrared images. It is motivated by the fact that variational techniques, such as those employed for computing the optical flow, are successfully applied to many computer vision applications but fail in this particular applicative context, mainly because optical flow techniques are adapted to rigid objects on visible data. The objective of this work is not to propose a full operational process for CMW estimation, but rather to improve optical flow techniques by applying constraints adapted to the specificity of meteorological infrared imagery.

1. Introduction

Satellite data are being more and more used for providing data to environmental models. Meteorology is undoubtfully the domain for which satellite data are most intensively used, but applications to oceanography, hydrology, air and water quality are gaining increased attention. Satellite data can be processed in many ways: the measures of electromagnetic radiations directly provide physical information, such as, in the context of atmospheric studies, brightness temperature on the infrared part of the spectrum, that can be assimilated into models. Further physical modelling makes it possible to derive optical depths that can be related to the chemical and aerosol content of the atmosphere. Many other physical interpretations of satellite measures are possible. Other interesting approaches consist of considering satellite data as images — three dimensional images in the case of image sequences — and applying methods initially developed for computer vision. These approaches allow for capturing the spatial and temporal behavior of macroscopic structures (e.g. clouds in meteorological imagery). However, the direct application of computer vision techniques to satellite data is often unsuccessful, since these methods, originally developed for specific imageries (video, medical imaging, …), do not account for the specificity of satellite images: nature of structures (e.g. clouds) and physical interpretation of pixels (e.g. temperature).

Cloud motion wind (CMW) estimation is a typical example of such methods. Many authors actually developed tracking techniques (e.g. statistical correlation as a tracking criterion) for that purpose. Another alternative is the use of variational techniques (e.g. optical flow), even though optical flow has not proven successful for cloud tracking. Correlation and variational methods have been used for long to analyse video image sequences. They cannot be used for operational CMW estimation in their original form. It is actually necessary to perform prior processing steps such as cloud height determination. Yet, the core of the tracking methods are similar to "traditional" image processing techniques. See the reference list in Velden et al. (1997) for a survey of techniques employed for CMW estimation.

The work presented therein describes a novel technique for CMW estimation on data provided by geostationary infrared radiometers, such as Meteosat's IR channel. The objective of this work is not to produce a full operational process (Laurent 1991, 1993) for deriving CMW from Meteosat data, but rather to improve the pure image processing part of these processes. We were particularly interested in improving the optical flow technique, that has had several successful applications in computer vision, but is rather disappointing for CMW estimation. The proposed scheme is a variational technique, similar to optical flow. Its major improvement lies

* Corresponding author. Tel.: +33 1 39 63 5703.
E-mail address: dominique.bereziat@inria.fr (D. Béréziat).

1364-8152/00/$ - see front matter
PII: S1364-8152(00)00046-3

in the fact that it does not make assumption on the behavior of individual pixels (e.g. brightness conservation), which usually does not make sense on meteorological images. In contrary, conservation constraints are applied to objects as a whole. This hypothesis, the total brightness conservation, leads to a differential constraint very similar to that of optical flow, with an additional divergence term. The latter allows for capturing expansion and contraction of deformable structures, as it is often encountered in turbulent flow. Beside this new formulation of an image processing problem, another physical justification can be found by analogy with fluid mechanics: the differential constraint is similar to the Euler equation of mass conservation. Furthermore, considering that the measured IR brightness temperature are highly correlated to cloud altitude, the differential constraint can be viewed as a cloud volume conservation hypothesis. The achieved results are much better than those obtained by applying optical flow techniques.

This paper is organized as follows: firstly, an overview of the image processing techniques used in meteorology for CMW estimation is given. The second part of the paper addresses the total brightness conservation hypothesis and how it leads to a new differential constraint. The hypothesis is justified from the point of view of image processing (conservation rules applied to objects rather than to pixels), and an analogy with fluid mechanics is given. A qualitative meteorological justification is also proposed. The third part gives details on the numerical scheme: even if a full operational process is not given, the reader can find all the necessary information for implementing the method. Finally results are displayed and compared to optical flow techniques.

2. State of the art

Understanding and measuring the dynamics of clouds is a complex issue. For instance, vortices are represented with chaotic models and fluid mechanics models. Such models are computationally expensive, leading to difficult problems in any real-time analysis procedure. Image analysis could be a very interesting way to analyse cloud motion because it only makes use of the images properties. A second motivation is related to the huge amount of digitalized data available, which rises an important need of automatic analysis of the data coming from the different satellites and sensors: geostationary satellites (MeteoSat, GMS, GOES, INSAT, GOMS) are providing for a long time a global coverage of the Earth's surface, with high temporal resolution (e.g. half-an-hourly) and a large variety of spectral measures (radiometers operating in the visible and IR domain). Currently, computer vision techniques are used by meteorologists to improve and analyze remote sensing data. There are two main approachs to compute velocity on these data. The first one are the correlation-based method. The second one are the optical flow-based method. In the following paragraphs, we describe and briefly confront these methods.

2.1. Correlation methods

Correlation methods are matching approachs: one tries to match a region of a frame in an image sequence to an another region on the next frame in the sequence using a correlation criterion. The correlation criterion measures the similarity between two regions. For a given region, the region on the next frame with maximal correlation is searched. In general, the correlation criterion used is the statistical correlation but others statistics may be used. It could be more interesting to use criteria modelling the clouds texture for example. This method is widely used on meterological data because it gives good results. Particularly, correlation methods are used to track small cloudy structures (the so-called *tracers*): the object to track has to have a constant size and a constant britghness. The correlation criterion identifies more easily small objects because they have a signifiant variance. The method has severals drawbacks:

- the computational cost is very expensive,
- the method is not usable on large structures,
- computation of a dense velocity field is impossible because the methods work only on objects, not on pixels,
- the method is very sensitive to the parameter (size of window computation) and requires, in order to speed up the computation, the prior localization of the tracers.

However, the correlation methods remain very widely used because they provide an excellent estimation of the objects' displacement.

2.2. Optical flow methods

The optical flow is the distribution of the apparent motion in an image. So, we compute directly the velocity and not the displacement as in the correlation methods. The fundamental assumption made is the brightness function, usually denoted I, is invariant for each point: given $(x,y)\in\Omega, I(t,x,y)=C\text{ste}$, $\forall t\geq 0$ where Ω is the image and t the time. This is formulated as $\mathrm{d}I/\mathrm{d}t=0$ and the chain rule derivation leads to the following equation, the *Optical Flow Constraint*:

$$\nabla I.w+I_t=0 \tag{1}$$

where w denotes the velocity vector and $I_t=\partial I/\partial t$. An important issue is that the Eq. (1), being underconstrained (one has to solve for w, that has two components, the two horizontal directions, with solely one

equation), cannot be solved directly. A second constraint is required. A simple approach consists of constraining the spatial variation of ∇w: it means that only regulars fields ($w \in L_2(\Omega)$) are allowed. The choice of the additional constraint depends on the context. In the meteorological framework, the method gives interesting results but could be improved by using a physical constraint. Many authors have undertaken studies on the choice of the second constraint for solving underconstrained problems in general (Black, 1994; Kornprobst et al., 1997) and more specifically the optical flow problem (Horn and Schunck, 1981; Nagel, 1983; Verri et al., 1990; Tistarelli, 1996; Herlin et al., 1998).

The optical flow can be seen as a matching method operating at the pixel level. Thus, a velocity is computed in each point of the image, providing a dense velocity field. A second advantage of the method is its low computational cost: the Horn et al. method takes about 2 min to compute a 256×256 velocity field image (including the necessary spatio-temporal gradient computation cost) on a DecAlpha 500 MHz. The method has however some inconvenients:

- the velocity estimation on the border of the image sequence is often unreliable: this problem is due to the difficulties in estimating the spatial gradient in these areas,
- if the velocity direction is generally well estimated, the velocity norm is under evaluated. This problem is due to the choice of the second constraint, a regularizing constraint which has no physical justification,
- the brightness invariance principle is not necessary true on meteorological data. For example, on infrared data, the temperature change over time. There are some alternative methods introducing a more flexible constraint than the brightness invariance (Gennert and Negahdaripour, 1987; Odobez and Bouthemy, 1995).

Since the optical flow methods provides a bad estimation of the velocity norm, they are seldom used for CMW estimation.

2.3. Conclusion and objectives

The two class of methods have their own advantages and drawbacks. The correlation-based methods provide a good estimation of velocity but at the expense of a high computational cost, and most implementations of correlation methods require a user supervision. In contrary, the optical flow methods are fully automatic, have a low computational cost but the quality of velocity estimation is unsufficient with respect to operational meteorological requirements. The goal of the next section is to propose an alternative method to the correlation-based methods and the optical flow-based methods: a non supervised method with a low computational cost and providing a realistic velocity field.

3. Total brigthness invariance hypothesis

3.1. Definition

The main idea of the method is to extend the brightness invariance principle from the pixel to the object. For that purpose we define the *total brightness* notion.

Definition 1 (Total brightness).

Let $\mathcal{O} \subset \Omega$ be an object, typically a cloud. The total brightness of $\mathcal{O}$ is defined by the quantity:

$$\mathcal{I}_{\mathcal{O}}(t) = \int_{\mathcal{O}} I(t,X,Y)\, \mathrm{d}X\, \mathrm{d}Y$$

where X and Y denote the spatial directions in the image domain.

Instead of assuming that a point has a constant brightness over time, we assume that a moving object point has locally a constant total brightness over time that is expressed by the relation:

$$\forall S \subset \mathcal{O},\ \frac{\mathrm{d}\mathcal{I}_S(t)}{\mathrm{d}t} = 0 \tag{2}$$

where S is any (topological opened) part of $\mathcal{O}$.

The total brightness invariance and the brightness invariance are not equivalent principles. However, in the case of a rigid object with a constant brightness, it verifies the total brightness invariance principle. This hypothesis may be viewed as an extension of the brightness invariance operating at an object scale.

Let $\varphi(t,x,y)$ be the spatial position at time t, the pixel being at the position (x,y) at time $t=0$. In fact, φ describes the trajectory of a point when t varies. Now, let $(X,Y)=\varphi(t,x,y)$ be the change of variables, the total brightness of a part S becomes:

$$\mathcal{I}_S(t) = \int_S I \circ \varphi(t,x,y) J_\varphi(t,x,y)\, \mathrm{d}x\, \mathrm{d}y \tag{3}$$

with J the Jacobian operator. We apply the chain rule derivation and obtain:

$$\forall S \subset \mathcal{O},\ \frac{\mathrm{d}\mathcal{I}_S(t)}{\mathrm{d}t} = \int_S \frac{\mathrm{d}}{\mathrm{d}t}[I \circ \varphi(t,x,y) J_\varphi(t,x,y)]\, \mathrm{d}x\, \mathrm{d}y \tag{4}$$

$$= \int_S \left[(\nabla I \circ \varphi).\varphi_t J_\varphi + I \circ \varphi \frac{\partial J_\varphi}{\partial t} \right] \mathrm{d}x\, \mathrm{d}y$$

where $\nabla = (\partial/\partial x, \partial/\partial y)$ and . is the scalar product in $\mathbb{R}^2$. We use the approximation $\mathrm{d}/\mathrm{d}t(J_\varphi) \sim \mathrm{div}(\varphi_t) J_\varphi$. and obtain

$$\forall S \subset \mathcal{O}, \int_S [\nabla I \circ \varphi . w + I_t \circ \varphi + I \circ \varphi \mathrm{div}(w)] J_\varphi \, \mathrm{d}x \, \mathrm{d}y. \quad (5)$$

Finally, Eq. (2) leads to:

$$\forall S \subset \mathcal{O}, \int_S (\nabla I . w + I_t + I \mathrm{div}(w)) J_\varphi \, \mathrm{d}x \, \mathrm{d}y = 0 \quad (6)$$

We can show that Eq. (6) is equivalent to:

$$\forall (x,y) \in \mathcal{O}, \nabla I . w + I_t + I \mathrm{div}(w) = 0. \quad (7)$$

3.2. Link with the extended optical flow constraint

Eq. (7) is known as the extended optical flow constraint. This model was originally studied by Schunck (1984). Eqs. (1) and (7) are very similar. Actually, Eq. (7) is exactly the mass conservation equation used in fluid mechanics where I and w are respectively the density and the velocity vector of a fluid parcel. The main problem with this equation is that there is no physical justification for using this model as a motion equation on images sequence. Horn et al. had removed the divergence term in Eq. (7) in order to interpret this model as a brightness invariance hypothesis. However, Schunck had proved the apparent motion computed with the Eq. (6) is equal to the real 3-D motion in the case of rotational motion in the direction of the camera axis. The optical flow Eq. (1) indeed cannot compute the real 3-D motion. Our work show now that the extended optical flow constraint has a physical interpretation: it is equivalent to a total brightness invariance hypothesis and can be used when this hypothesis is verified.

3.3. Physical interpretation of the total brightness invariance

In the previous section, we demonstrated that the total brightness invariance could be justified in the context of computer vision and have have shown an analogy with mass conservation in fluid mechanics. We now try to find a justification in the specific context of meteorological IR data.

The first step is based on the physical interpretation of the measured radiance. The data consist of digital counts that can be converted into brightness temperatures by applying the calibration procedures, provided simultaneously with satellite data. Considering the qualitative description of the vertical profile of temperature in the troposphere (i.e. a negative gradient of 6 degrees per kilometer), we see that brightness temperatures are somehow negatively correlated to cloud elevation. This is the core of our "justification", and this in fact relies on two major simplifications: the first one is that this assertion is true if the structures of interest are below the altitude of temperature inversion, which excludes the polar stratospheric clouds from the study. However, as we are mainly concerned with large vortex structures in northern Atlantic, this hypothesis is valid. We therefore consider that the temperature profile is in the form $T(z)=T_0-\alpha z$. The second simplification consists of considering that radiances measured on clouds correspond to emissions at cloud top, instead of the altitude for which the atmosphere's optical depth equals 1. This is equivalent to state that the clouds are opaque to infrared radiation and that the measured radiance is emitted by the cloud, which is not an irrealistic assumption when considering very large and deep cloud structures.

Stating this, we now consider that the radiance is a measure of elevation. It is easy to see that the total brightness of a cloud equals (up to a linear transformation) the volume of the column of air located between an altitude of reference and the cloud top. The next level of approximation is the assimilation of the variation of this volume to the variation of the cloud volume. This is a rather severe assumption on the deformation that clouds undergo during their motion, our belief being that, at least during short periods and if there is no precipitations, this can be done. The final conclusion is that the total brightness invariance is equivalent to the conservation of the cloud volume. This of course has to be considered as a qualitative meteorological justification, and does not pretend to exactitude.

3.4. Bi-model

In the subsection 3, we've proved that the total brightness hypothesis [Eq. (2)] is equivalent to Eq. (7). Strictly speaking, this equivalence is true only on some regions of an image. For the meteorological infrared data example, only clouds should verify the total brightness invariance hypothesis. We use the fact that the Eqs. (1) and (7) are very similar: the second one contains an additional term: $I\mathrm{div}(w)$. Strictly, Eq. (7) is applied on points $(x,y) \in \mathcal{O}$. Outside $\mathcal{O}$, we propose to use a standard brightness invariance assumption to model the motion. Then we obtain a new motion equation that is a bi-model:

$$\forall (x,y) \in \Omega, \nabla I . w + I_t + I \mathrm{div}(w) \mathbb{1}_{\mathcal{O}} = 0. \quad (8)$$

4. Numerical resolution

We present in this section an implementation of the numerical resolution of Eq. (8). The first problem is to choose a second constraint to -overconstraint Eq. (8) (w has two components). We choose to regularize the spa-

tial variation of w using a $L_2(\Omega)$ norm. Of course, this choice is not really pertinent in our framework but it gives satisfying results. A variational framework is used: we build a functional which minimum value provides a solution for the Eq. (8). This functional is defined by:

$$E(w)=\int_{\Omega}(\nabla I.w+I_t+I\mathrm{div}(w)\mathbb{1}_{\mathcal{O}})^2\,\mathrm{d}x\,\mathrm{d}y+\alpha\int_{\Omega}(\|\nabla u\|^2+\|\nabla v\|^2)\,\mathrm{d}x\,\mathrm{d}y \quad (9)$$

where (u,v) = w and $\|\nabla u\|^2=u_x^2+u_y^2$. The first integral is the bi-model constraint, the second one is the regularizing term. The parameter α weights the effects of the regularizing constraint. We can change this empirical value to obtain velocity fields more and less smoothed. We have to compute the minimul value of E. The operand w is a function lying in $H_1(\Omega)$. A minimum of E can be computed by using the Euler Lagrange equations associated to (9). The Euler Lagrange equations are the differential of E with respect to w and in the case of (9) equal:

$$\begin{cases}-\dfrac{\partial}{\partial x}(I(I_xu+I_yv+I_t+(Iu_x+Iv_y)\mathbb{1}_{\mathcal{O}}))-\alpha\Delta u=0\\ -\dfrac{\partial}{\partial y}(I(I_xu+I_yv+I_t+(Iu_x+Iv_y)\mathbb{1}_{\mathcal{O}}))-\alpha\Delta v=0\end{cases}. \quad (10)$$

As the function E is convex, the Eq. (10) provide an unique minimal value of (9). Such a system is difficult to solve. If we suppose, in a first time, that $\mathbb{1}_{\mathcal{O}}\equiv 1$, Eq. (8) is equivalent to:

$$\mathrm{div}(W)+I_t=0 \quad (11)$$

where $W=Iw$. The functional to minimize becomes:

$$E(W)=\int_{\Omega}(\mathrm{div}(W)+I_t)^2\,\mathrm{d}x\,\mathrm{d}y+\alpha\int_{\Omega}(\|\nabla U\|^2+\|\nabla V\|^2)\,\mathrm{d}x\,\mathrm{d}y \quad (12)$$

where $(U,V)=W$. The Euler–Lagrange equations associated to E are:

$$\begin{cases}-\dfrac{\partial}{\partial x}(I_t+U_x+V_y)-\alpha\Delta U=0\\ -\dfrac{\partial}{\partial y}(I_t+U_x+V_y)-\alpha\Delta V=0\end{cases}. \quad (13)$$

The resolution of Eq. (13) are performed by using the numerical scheme:

$$\begin{cases}\dfrac{\mathrm{d}U}{\mathrm{d}\tau}-\dfrac{\partial}{\partial x}(I_t+U_x+V_y)-\alpha\Delta U=0\\ \dfrac{\mathrm{d}V}{\mathrm{d}\tau}-\dfrac{\partial}{\partial y}(I_t+U_x+V_y)-\alpha\Delta V=0\end{cases}. \quad (14)$$

We can show that a stationary solution of Eq. (14) is solution of Eq. (13). Finally this scheme is implemented using the finite difference method: $\mathrm{d}W/\mathrm{d}\tau\sim W^{\tau+1}-W^{\tau}$. We have:

$$\begin{cases}U^{\tau+1}=U^{\tau}+\dfrac{\partial}{\partial x}(I_t+U_x^{\tau}+V_y^{\tau})+\alpha\Delta U^{\tau}\\ V^{\tau+1}V^{\tau}+\dfrac{\partial}{\partial y}(I_t+U_x^{\tau}+V_y^{\tau})+\alpha\Delta V^{\tau}\end{cases}. \quad (15)$$

At the first step, we suppose $U^0=V^0=0$. Others differential operators on W are approximated by finite differentiation. We have:

$$\begin{cases}U_{xx}(i,j)\sim(U(t,i+1,j)-2U(t,i,j)+U(t,i-1,j))/4\\ U_{yy}(i,j)\sim(U(t,i,j+1)-2U(t,i,j)+U(t,i,j-1))/4\\ U_{xy}(i,j)\sim(U(t,i+1,j+1)-U(t,i-1,j)+U(t,i,j-1)\\ -U(t,i-1,j-1))/4\end{cases}. \quad (16)$$

The computation of the scheme (14) provides a dense velocity vector field on an object O following the extended optical flow constraint. Outside $\mathcal{O}$, the divergence term disappears. The functional to minimize is:

$$E(w)=\int_{\Omega}(\nabla I.w+I_t)^2\,\mathrm{d}x\,\mathrm{d}y+\alpha\int_{\Omega}(\|\nabla u\|^2+\|\nabla v\|^2)\,\mathrm{d}x\,\mathrm{d}y \quad (17)$$

the Euler–Lagrange equations are:

$$\begin{cases}-I_x(I_xu+I_yv+I_t)-\alpha\Delta u=0\\ -I_y(I_xu+I_yv+I_t)-\alpha\Delta v=0\end{cases} \quad (18)$$

and the numerical scheme is:

$$\begin{cases}u^{\tau+1}=u^{\tau}+I_x(I_xu^{\tau}+I_yv^{\tau}+I_t)+\alpha\Delta u^{\tau}\\ v^{\tau+1}=v^{\tau}+I_y(I_xu^{\tau}+I_yv^{\tau}+I_t)+\alpha\Delta v^{\tau}\end{cases} \quad (19)$$

this scheme provides a dense velocity vector field outside the object $\mathcal{O}$ following the optical flow constraint. Horn et al. had used a different numerical scheme to compute the solution of Eq. (18), that seems to be more robust than ours, but it can not be used in the extended optical flow constraint. The choice of the parameter is empirical. We use the value α=40.

5. Results

To validate the total brightness conservation model, we built a synthetic sequence verifying this principle: on Fig. 1, the grey square evolving over time keeps a constant total brightness. The model correctly captures the motion.

Fig. 2 depicts the result of the brightness invariance

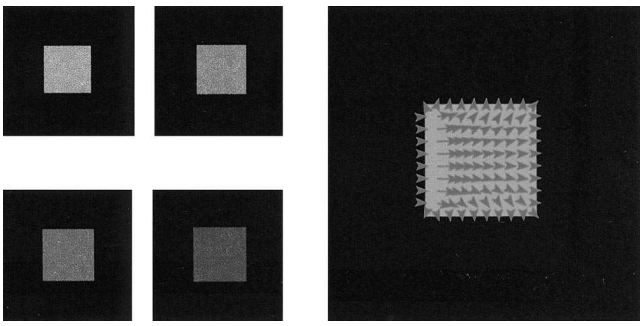


Fig. 1. Sequence of a growing square with a constant total brightness and an horizontal translational motion (from left to right). On left, the four first images of the sequence. On right, the result of the computation of velocity field.

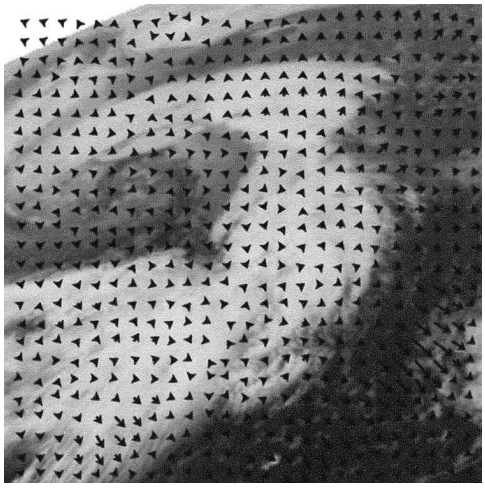


Fig. 2. Result of the brightness invariance model on an IR image.

model (the optical flow constraint) on a METEOSAT IR sequence of a vortex over the Atlantic Ocean. There are many differences between the optical flow and the extended optical flow: they are underlined at the end of this section. Differences between the extended optical flow model and the bi-model are not significative. The velocity field is different only outside of clouds because of the different motion models used.

To apply the bi-model model on our data, we have to localize cloudy structures on infrared data. It can be performed using a simple threshold on grey level value. However, we've used a sophificated segmentation process to localize accurately the cloudy structures. Fig. 3 pictures the results of the total brightness model (the extended optical flow constraint) and the bi-model.

Figs. 4 and 5 picture respectively the results of the brightness invariance model on two images of the vortex sequence and the total brightness invariance model. We used the Vis5D software to display the vector velocity field and the iso-value of the norm of the field. Comparing the brightness invariance hypothesis with the total brightness invariance hypothesis, we have the following remarks:

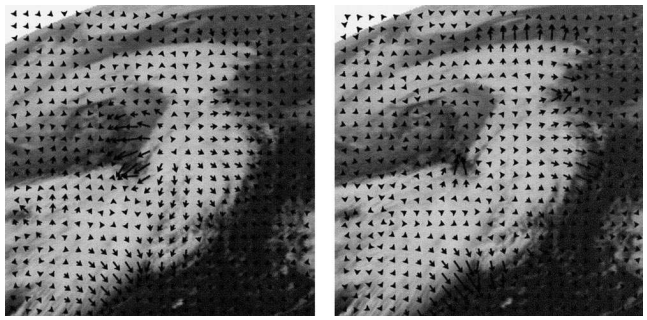


Fig. 3. Result of the total brightness invariance model (left) and the bi-model (right).

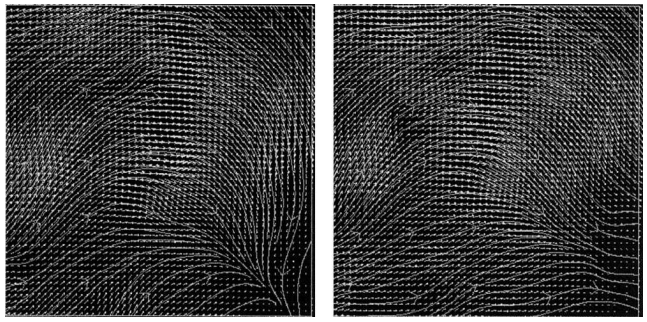


Fig. 4. Brightness invariance model computed on three successive images.

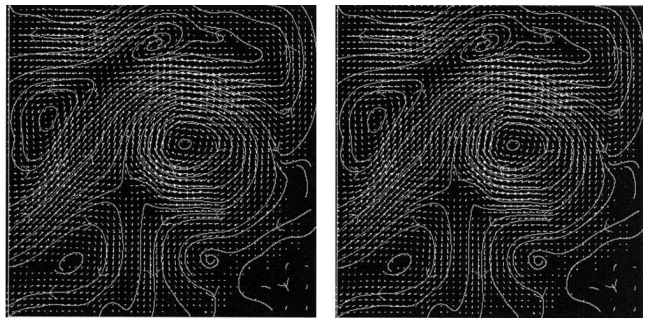


Fig. 5. Total brightness invariance model computed on three successive images.

- detection of singular points: our method captures several singular points (see Fig. 5). The classical optical flow model does not,
- the optical flow method detect motion in darker areas, i.e. areas where there is no cloud and then there is no motion. Our model does not detect any motion in these areas because their grey level values are close to zero,
- our model detects a large motion activity at the border of the vortex, larger than motion computed with the optical flow method. Local expansion/contraction are captured with our model due to the divergence term modeling the type of motion.

6. Conclusion

In order to improve the results of optical flow techniques applied to cloud motion wind estimation, a novel hypothesis, the total brightness conservation is proposed. This hypothesis leads to a differential equation similar to the mass conservation equation in fluid mechanics. A qualitative meteorological justification can be found if one considers that the radiance measured by infrared radiometers is directly related to elevation. This of course is not exactly the case, and a prior cloud heigh resolution should be performed first before applying our method. However, the results show significant improvements compared to optical flow, and it can be expected that the integration of this algorithm into an operational CMW estimation chain might be beneficial. This in fact is the principal perspective of this work, that has to be undertaken in order to perform a true meteorological validation of the results.

References

Black, M., 1994. Recursive non-linear estimation of discontinuous flow fields. European Conference on Computer Vision 2, 238–245.

Gennert, M., Negahdaripour, S., 1987. Relaxing the brightness constancy assumption in optical flow. Technical report 975. MIT AI Lab.

Herlin, I., Cohen, I., Béréziat, D., 1998. Wind estimation by image processing for air pollution modelling. SAMS 32, 57–66.

Horn, B.K.P., Schunck, B.G., 1981. Determining optical flow. Artif. Intell. 17, 185–203.

Kornprobst, P., Deriche, R., Gilles, A., 1997. Non linear operators in image restoration. In: CVPR97, pp. 325-330.

Laurent, H., 1991. Wind extraction from multiple meteosat channels. In: Proc. Workshop on Wind Extraction from Operational Meteorological Satellite Data, Darmstadt.

Laurent, H., 1993. Wind extraction from meteosat water vapor channel image data. J. Appl. Met. 32, 1124–1133.

Nagel, H.H., 1983. Displacement vectors derived from second-order intensity variations in image sequences. Comput. Graph. Image Process. 21, 85–117.

Odobez, J.M., Bouthemy, P., 1995. Robust multiresolution of parametric motion models. In: JVCIR, vol. 6, pp. 222-238.

Schunck, B., 1984. The motion constraint equation for optical flow. In: Proc. 7th ICPR, Montreal, pp. 20-22.

Tistarelli, M., 1996. Multiple constraints to compute optical flow. IEEE Trans. Pattern Anal. Mach. Intell. 18, 1243–1250.

Velden, C.S., Hayden, C.M., Nieman, S., Wenzel, W.P., Wanzong, S., Goerss, J., 1997. Upper-tropospheric winds derived from geostationary satellite water vapor observations. Bull. Am. Meteor. Soc. 78, 173–195.

Verri, A., Girosi, F., Torre, V., 1990. Differential techniques for optical flow. J. Opt. Soc. Am. 7, 912–922.

A.2 Traitement d'images dans le contexte sphérique

[1] W. Rekik, D. Béréziat, and S. Dubuisson. Image processing in spheric-shaped data using a geographical transformation. application to bio-cellular imaging. In *International Symposium on Biomedical Imaging (ISBI)*, pages 1256–1259, Virginia (USA), April 2006. IEEE.

[2] W. Rekik, D. Béréziat, and S. Dubuisson. Optical flow computation and visualization in spherical context. application on 3D+t bio-cellular sequences. In *Engineering in Medicine and Biology Society (EMBS)*, pages 1645–1648, New-York (USA), September 2006. IEEE.

IMAGE PROCESSING IN SPHERIC-SHAPED DATA USING A GEOGRAPHICAL TRANSFORMATION. APPLICATION TO BIO-CELLULAR IMAGING.

Wafa Rekik, Dominique Béréziat and Séverine Dubuisson

Université Pierre et Marie Curie
Laboratoire d'informatique de Paris 6
8, rue du Capitaine Scott
75015 Paris, FRANCE
Wafa.Rekik@lip6.fr.

ABSTRACT

In this article, we put forth the contribution of cartographic map projection in image analysis for $2D$ and $3D$ spheric-shaped data. In the $2D$ case, map projection of the underlying hemi-spherical surface has the merit to overcome deformations generated by perspective sights. In the $3D$ case, processing cartographic references of the original spheroid surfaces presents mainly two advantages. On one hand, it decreases significantly the amount of processed data. On the other hand, we avoid extension of exhaustive processes to a higher dimension. We make use of this geographical transformation in several classical image processing applications such as temporal tracking, information visualization, and feature extraction, such as segmentation. These applications are described and results of cell wall simulations are displayed.

1. INTRODUCTION

In bio-cellular imaging, microscopic cell wall models are designed in order to analyze the behavior of some biological structures and find out their functional role. Due to some biological constraints, these models have spheroid shapes. Biologists inject fluorescent markers on the spheroid surface, in order to segregate between objects interacting with the input light spot. Multidimensional types of cell wall simulations are acquired: temporal 2D video sequences, *i.e. 2D+t*, multi-focal 2D sequences, *etc*. We may merely distinguish two types of acquisitions: *2D* and *3D* frames. In the 2D case, frames are perspective projections of light signals emitted by structures of interest lying on the translucent spheroid surfaces. Distortions introduced by projective sights of these non-rigid objects generate deformations on their shape, size, *etc*. These deformations bother the extraction of typical features and relevant information related to structures of interest, namely, temporal tracking of such templates may fail. Consequently, it is an essential issue to take into account the underlying hemi-spherical surface on which these structures are evolving, without modifying the topology of the input data. For this purpose, we first warp the 2D frames into a spherical image and then project the 3D surface onto a planar cartographic reference using a suitable map projection. This geometrical transformation locally removes distortions, generated by perspective projection, around the objects of interest. It is then possible to carry out different analysis in transformed frames. In the 3D case, we have obviously accurate description of the original spherical surface, free of any projective sight distortions. A natural first approach consist in directly processing the 3D data sets. This way of proceeding is doubtless reasonable. However, extension of non trivial algorithms for *2D* to *3D* domains appear to be quite exhaustive. On one hand, the amount of processed data increases significantly, which bring algorithms about being computationally prohibitive. On the other hand, some basic applications doesn't perform well, like optical flow computation. We may also think of statistical local approaches, such as Markov Random Fields modelisation, that involves specific neighborhoods. In the 3D case, voxels surrounding a selected point seem to form a bent vicinity. It is then interesting to consider the cartographic reference of the corresponding 3D images. It is worth noticing that, in this case, the $3D$ elliptical surface is already available, thereby, there is no need to map texture into a sphere. We proceed with the description of the geographical transformation in section 2. Since the map projection involved generates a set of scattered points, we present in section 3 a method for missing-value approximation. Then, we present main applications of our approach: temporal tracking in section 4.2 as well as information visualization in section 4.3 and extraction in 4.4.

2. GEOGRAPHICAL TRANSFORMATION

Our geographical transformation is based on a suitable cartographic map projection. The tangent map projections are divided into three categories, depending on the choice of the surface on which the Earth surface will be unrolled: conical, cylindrical and azimuthal. The first two are purely geometri-

cal projections of the Earth upon a tangent cone or cylinder. The parallel on contact with the surface is mapped without any deformation and corresponds to the line of zero distortion on the map. Azimuthal projections project the globe onto a plane tangent to the Earth's surface on a specified point. Distances measured from the center are true, and distortions of other properties increase away from this center point. In this case, the point of contact coincides with the point of zero-distortion on the map, that is very interesting if we need to minimize distortions around a specific region of interest. Since we deal with objects of weak thickness lying on the surface, we opt for the *azimuthal equal area projection*. The latter maps locally the region of interest and limits deformations around it.

In the $2D$ case, we observe circular contours of the surface of interest. We estimate then maximum of curvature by recovering these contours. They are fitted by the closest tangent circle whose parameters correspond, with extension to a $3D$ representation, to the hemispherical underling surface ones. Once these parameters are estimated, we map the $2D$ texture into the spherical surface and then compute the azimuthal equal area projection (see [1, 2] for a complete mathematical analyze) of the mapped surface. These steps are schemed in Figure 1. In the $3D$ case, there is obviously no need to warp the texture since the spheroid surface is already available.

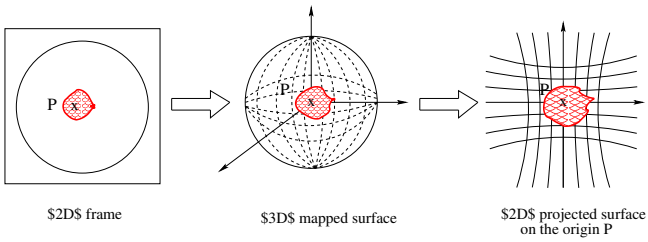


Fig. 1. From $2D$ frame to planar cartographic reference.

It worth noticing that map projection result depends on the accuracy of the original curvature estimation. For this purpose, we first detect the edges in the image using a Canny-Deriche operator to obtain a transition map. Then, we use an edge following algorithm to detect the longest contours of the transition map, corresponding to the contours of the surface on which objects of interest are lying. These contours are fitted by a tangent circle whose parameters (center and radius) are computed by least square fitting. Moreover, couples of computed cartesian coordinates of projected points hardly correspond to a grid position in the sampled output image. In addition, geometrical distortions introduced by this projection beyond the close neighborhood of the origin, generate important number of non-initialized points on the mapped surfaces. That's why we used an interpolation method in order to estimate values of missing points.

3. MISSING-VALUE APPROXIMATION

Our projection yields a set of scattered data in the projection plan. The farther we move from the projection origin, the more mapped samples become sparse and irregular. Our goal is to fit a smooth surface through a non uniform distribution of data samples. For this purpose, we manage to represent our surface as a sum of weighted and shifted synthesis functions. B-splines stand apart for a good compromise between quality (namely high fidelity reconstruction and regularity) and computational issues.There is a whole family of synthesis kernels made of B-splines, usually called β^n, as they are $(n-1)$ continuously differentiable. They are given by :

$$\beta^o(x) = \begin{cases} 1 & |x| < \frac{1}{2} \\ \frac{1}{2} & |x| = \frac{1}{2} \\ 0 & |x| > \frac{1}{2} \end{cases} \quad \text{and} \quad \beta^{n+1}(\mathbf{x}) = \beta^n(\mathbf{x})\beta^0(\mathbf{x})$$

Since our samples are scattered and mapped to non-grid position, approximation with such functions is a non trivial task. A framework introduced in [3], is dedicated to this sake. The implemented algorithm makes use of a coarse-to-fine hierarchy of control lattices in order to generate a sequence of bi-cubic B-spline functions whose sum approaches the desire interpolation function. We carry out this method allowing a high quality of approximation with regard to a satisfactory computation time.

4. APPLICATIONS

4.1. Biological context

We deal with spatio-temporal video sequences of cell wall simulations acquired by a CCD camera coupled with a microscope. Recall that camera and microscope observe light signals emitted by some fluorescent markers injected on the cell wall surface modeled by ellipsoid vesicles. These markers target at viewing some lipid exterior phases of weak thickness that we call *rafts*, moving on these surfaces. Biologists are interested on analyzing behavior of rafts (shape, movement, *etc*.) in order to find out their functional role (see [4, 5] for details). Due to this experimental context, different acquisitions are blurred and have a very low contrast resolution. That doesn't ease application of image processing algorithms.

4.2. Temporal tracking

Many methods developed to track objects in motion make use of correlation techniques, consisting of two steps: first, to pick up suitable features in the image, second, to search the corresponding points in another image to determine their displacement. It is the case for Block Matching methods, interesting candidates to analyze motion of quite small object structures included within a window defining the block considered. So, if we consider two consecutive frames I_1 and I_2,

such approach attempt to compute ξ so that:

$$\xi = \underset{\overrightarrow{T}}{\operatorname{argmax}}\, S(I_1|_{w_1}, I_2|_{w_1+\overrightarrow{T}})$$

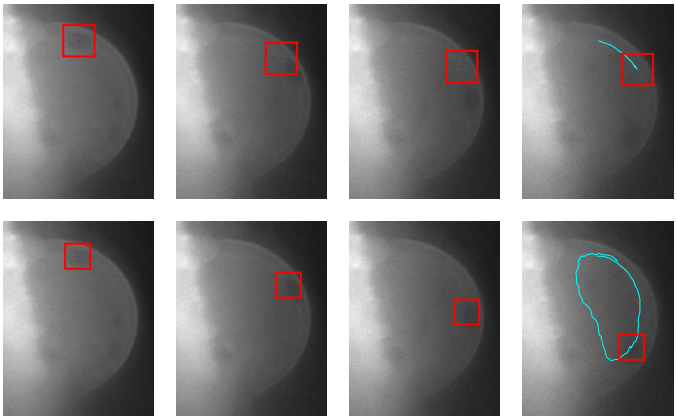
S is a similarity criterion, in our case it is the well-known normalized correlation coefficient. Let $I_1|_{w_1}$ be the restriction of I_1 to the window w_1 and $\overrightarrow{T}$ the tested displacement vector. Indeed, to find a characteristic feature from the first image I_1 in the second I_2, we take a small area (*i.e.* window) w_1 from the first image and compare it with equal-sized areas $w_2 = w_1 + \overrightarrow{T}$ from the second image within a certain range (*i.e.* search space). In this search space, we look for the position of the optimum similarity between the two areas, giving a displacement vector of the characteristic feature between the two images. The idea is to combine the Block Matching tracking with the geometrical transformation realized with map projection (section 2). That means that, before estimating correlation between two blocks w_1 and w_2, for each tested displacement vector $\overrightarrow{T}$, we compute map projections of these blocks taking as origin their centers. We note π_P map projection of a block w_1 taking as origin its center P. We estimate correlation then by the formula:

$$\xi = \underset{\overrightarrow{T}}{\operatorname{argmax}}\, S(\pi_P(I_1|_{w_1}), \pi_{P+\overrightarrow{T}}(I_2|_{w_1+\overrightarrow{T}}))$$

Consequently, correlation is computed between blocks which doesn't suffer under perspective projection deformations. Corresponding algorithm is described with details in [6].
We have tested our method on several video sequences and compared it with the standard Block-Matching method. The latter, *ie*, the correlation method fails quickly in image sequences, especially when the appearance of the *raft* is distorted because of the 2D view of its displacement on the vesicle surface. Comparisons between the two method results, on the same video sequence, are given in Figure 2. We can notice that, at the beginning of the sequence, windows localized by both methods have almost same center positions. Difference of quality tracking results are notable from the 16^{th} frame of the sequence (see Figure 2b). Moreover, we can see that the correlation method totally fails on the 22^{th} frame of the sequence, whereas our approach continues to capture *raft* motion (Figure 2c). Since the target is completely lost by conventional method, Block Matching is let down in the 23^{th} frame. However, tracking is carried out in the same sequence with our method till the end of the sequence. Figure 2d shows, in the 33^{th} frame, the tracking window, as well as, the total *raft* trajectory computed by both methods. In a general way, we have noticed, by testing other video sequences, that our approach prevents tracking errors, especially when the moving object appearance is deformed by 2-D acquisition.

4.3. Information visualization

We aim at visualizing structures of small thickness in $2D$ and $3D$ images. Our geographical transformation unrolls locally



(a) Frame 1 (b) Frame 16 (c) Frame 22 (d) Frame 33

Fig. 2. Comparison of the tracking results obtained by, from top to bottom, standard BM method and ours.

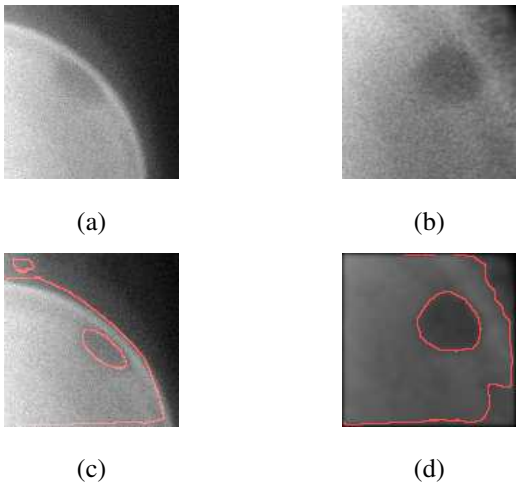
the spheroid surface around the object of interest. It presents then the advantage to improve the visualization effects. In the $2D$ case, deformations resulting from this kind of projection increase the recovery and the understanding on the visualized area. In fact, unlike perspective projection, this geographical transformation does not map uniformly each point of the 3D warped surface. As for the $3D$ case, our image is a structure of data organized in a certain number of equal-sized plans. Indeed, they have all the same number of lines and columns. Therefore, it is possible to visualize the sphere either, by its perspective projection, or as a sequence of 2D plans. The first solution leads to the same problem as in the $2D$ case, and the second one is insufficient to recover and display structures of small thickness lying on the spheroidal surface. The first couple of image in Figure 3 illustrates difference of visualization effects between both perspective and map projections. Moreover, varying the projection origin around the surface allows to observe different views of the sphere. We designed then a novel tool called Mapvis for the visualization of scalar and vectorial information lying on spheroidal surface, reader is referred to [7] for a full principle and illustration description.

4.4. Information extraction

Segmentation is an essential issue for relevant feature extraction such as surface shape and specific area perimeters. However carrying out segmentation on projected textures seems to be interesting for the $3D$ as well as the $2D$ case. On one hand, processing projected spheres decreases the dimension of the implemented algorithms which are, doubtless, computationally prohibitive when extended to the $3D$ case. Moreover in a region close to the projection center, original characteristics of the underling sphere are accurately preserved. On the other hand, in the $2D$ context, perspective projection generates de-

formed curvatures. The latter yields errors in surfaces and perimeter analyzing and bother matching algorithms based on curvatures.

Addressing the segmentation application, we have used a geodesic snake based method. A geodesic snake is an implicit formulation of a standard snake [8] using an Eulerian formulation.The main advantage of using an implicit snake is the capability to change the topology of curve. We experiment segmentation using the Chan&Vese model [9] involving a criterion of a gray level homogeneous repartition inside the curve. Moreover, data should be preprocessed otherwise the segmentation process failed. This problem is due to the low contrast of the data set. We enhance the image contrast by the mean of a two-step filtering process. In a first stage, contrast is enhanced using a combination of mathematical morphological operators. In the second one, we apply a numerical scheme of an anisotropic diffusion [10], which has the property to smooth image while preserving edges. Segmentation results are displayed in Figure 3: (a) is a perspective view of the $3D$ processed surface, (b) is the map projection of the selected volume, (c) is the initialization curve on the mapped surface and finally (d) is segmentation results. We can see that the geographical transformation improves the *raft* shape and contours visibility.



(a) (b)

(c) (d)

Fig. 3. Segmentation of geographically transformed spherical surface.

5. CONCLUSION

In this article, we have presented a new approach for processing spheric-shaped data. It involves a geographical transformation based on a suitable map projection. This transformation presents several advantages for the $2D$ as well as the $3D$ case. It improves temporal tracking of biological structures in $2D + t$ video sequences. Besides, it ameliorates recovery and visualization effects of scalar and vectorial information lying on spheroid surface. Moreover, feature extraction is more meaningful in geographically transformed $2D$ and $3D$ images. In other words, this transformation provides a cartographic reference of the original data with relevant properties improving a number of non-trivial image analysis. It is dedicated mainly to a spheroid geometry data context, but it could be interesting to generalize this concept to any regular surface. A solution could consist in estimating a global curvature of the enclosing surface or a local one near the object of interest.

6. REFERENCES

[1] Illife Jonathan, *Datums and map projections for remote sensing, GIS, and surveying*, Latheronwheel, caithness: Whittles Pub. Boca Raton: CRc press, c200.

[2] D.H Maling, *Coordinate Systems And Map Projections*, Pergamon Press, 2nd edition.

[3] Seungyong Lee, George Wolberg, and Sung Yong Shin, "Scattered data interpolation with multilevel b-splines," *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 3, pp. 228–244, July-September 1997.

[4] D.A Brown and A.E London, "Structure and origin of ordered lipid domain in biological membranes," *J. Membr. Biol.*, vol. 164, pp. 103–114, 1998.

[5] K. Simons and E. Ikonen, "Functional rafts in cell membranes," *Nature*, vol. 387, pp. 569–572, 1998.

[6] W. Rekik, D. Béréziat, and S. Dubuisson, "Tracking of non-rigid structures evolving on 3d surfaces by compensating perspective view deformations.," in *In the proceedings of IASTED-SIP*, Honolulu, Hawaii, USA, August, 15-17 2004.

[7] W. Rekik, D. Béréziat, and S. Dubuisson, "Mapvis: a map-projection based tool for visualizing scalar and vectorial information lying on spheroidal surfaces," in *In the proceedings of international conference of information visualization, IVO5*, London, the United Kingdom, July 2005.

[8] M. Kass, A. Witkin, and T. Terzopoulous, "Snakes : Active contour models," *International Journal of Computer Vision*, pp. 321–331, 1988.

[9] Tony F. Chan and Luminita A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 7, February 2001.

[10] Pietro Perona and Jitendra Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, July 1990.

Optical flow computation and visualization in spherical context. Application to 3D+t bio-celluar temporal sequences.

Wafa Rekik, Dominique Béréziat and Séverine Dubuisson
Université Pierre et Marie Curie, Laboratoire d'informatique de Paris 6
8, rue du Capitaine Scott,75015 Paris, France.
Wafa.Rekik@lip6.fr

***Abstract*— The main focus of this paper is optical flow computation for 2D movement of objects embedded on 3D surfaces. For spheric-shaped supports, object motion has only 2 degrees of liberty, thus 3D optical flow constraint is not relevant. Constancy assumption is formulated using a suitable parametrization of the 3D surface, leading to a 2D equation. Input temporal sequence is also transformed according to the 3D surface parametrization. We build a complete 2D model, taking into account the underlying spherical surface. It has the merit to estimate at a lower cost velocity field in the temporal input sequence. In order to analyze motion computation results, we design an adapted visualization tool, instead of carrying out an inverse transformation for the velocity field. Adapted to the selected parametrization, it displays rapidly moving objects and velocity field and improves the understanding of the displayed information. We display optical flow computation results for 3D+t cell wall simulation temporal sequences.**

I. INTRODUCTION

Motion computation is of growing interest in various computer vision applications. A fundamental widely investigated issue is the computation of optical flow. It aims at providing an approximation of the local image motion based upon local derivatives in a given sequence of images. We are mainly interested in estimating the velocity vector field of objects embedded on spheroid surfaces. We have therefore, a $3D + t$ structure of data organized on a set of 3D frames at each sample of the acquisition temporal process. Each 3D image is a ball represented by a given number of equal-sized plans. Since inside objects are meaningless, movement of structures evolving on the surface has only two degrees of liberty. 3D optical flow constraint is not relevant for 2D object movement. It is then useful to investigate motion on the surface domain that profit from a 2D parametrization. We switch then to a system of spherical coordinates, with strong resemblance to geographical one. We evaluate angular displacements, as radial temporal variations are negligible. The main advantage of this way of proceeding is to decrease the dimension of the input data set. We process then $2D + t$ instead of $3D + t$ sequences. For this sake, we need to map the 3D cartesian grids into an angular one. Due to angular distortion, a robust interpolation method is needed to recover missing points, looking like furrows. After the angular temporal variation estimation, it is possible to come back to the 3D structure of data. However, we are interested in visualizing, both texture and vector fields in the same reference. Since the geometry of the surface is not relevant, standard volume visualization methods do not seem to be adapted to our context of study. We developed a visualization tool based on a suitable map projection, called **Mapvis**, that maps the data set sampled on a 2D angular grid into a cartographic 2D reference. **Mapvis** has hence the merit to provide relevant visualization properties for scalar and vectorial information for 3D surfaces.

II. OPTICAL FLOW COMPUTATION

The initial hypothesis in measuring image motion is that the intensity structures of local time-varying image regions are approximately constant under motion for a short duration. The optical flow constraint, introduced by Horn and Shunck [1], expresses the assumption that a moving pixel keeps the same gray value over time. Formally, if $I(x, y, t)$ is the image intensity function, then:

$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t),$$

where δx and δy are the displacements of the local image region at (x, y, t) after time δt. Expanding the left-hand side of this equation in a first order Taylor series leads to the well known Optical Flow Constraint:

$$\overrightarrow{\nabla} I . \overrightarrow{\omega} + I_t = 0, \tag{1}$$

where $\overrightarrow{\nabla I} = (I_x, I_y)$ is the spatial intensity gradient and $\overrightarrow{\omega} = (\frac{\partial x}{\partial t}, \frac{\partial y}{\partial t})$ is the velocity vector. Many techniques were developed to solve equation 1. We may distinguish differential methods involving variational formulation, Markov random field approach [2] and parametric modeling [3], [4]. Differential methods include mainly local and global approaches. Local methods use normal velocity information in local neighborhoods to perform a least square minimization in order to find out the best fit for $\overrightarrow{w}$. That was first described by Lucas and Kanade [5]. Global methods use an additional global constraint, usually a smoothness regularization term, to compute dense optical flow over large image regions. Computation in the 3D domain is most frequently performed by extension of the last two techniques to a higher dimension. However, this solution is not adapted to our context of study. Each 3D image is an empty ball organized on a sequence of equal sized plans. That means that our 3D data set is merely a sequence of variable diameter rings of low thickness. Therefore, the optical flow constraint becomes not relevant

for such structure of data. Moreover, computation of spatial derivatives in the three directions, x, y and z is incoherent and noisy. An attempt to improve the 3D variational model is to penalizes radial component of the vector field. However, this solution increases the complexity of the model, already extended to a higher dimension. Therefore, we thought of a new approach consisting in carrying out estimation of the global and local spherical vector field.

III. Angular velocity computation

A. 3D dataset transformation

Since movement of structures embedded on spherical surfaces has only two degrees of liberty, a system of 2D coordinates is sufficient to parametrize the displacement vector field. Spherical angular coordinates seem well adapted to the geometry of the input dataset. The idea is to map the 3D$+t$ texture into the 2D$+t$ angular grid, *ie* indexed by θ, φ and t. For an entire globe, with reference to the geographical north pole and the equator, θ has range from 0 to π versus a double-sized range, for φ, *ie* from $-\frac{\pi}{2}$ to $\frac{3\pi}{2}$. The main limit of this coordinate system is to the singularity of geographical poles. A solution to avoid computation of velocity vectors close the geographical polar zones when we ought to, could be a globe rotation, see [6] for mathematical formulates. After localizing a viewpoint, we carry out a rotation of angular coordinates (θ, φ), with reference to this viewpoint and a great circle oblique to the equator (see Figure 1).

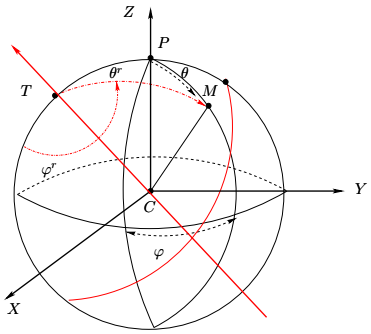


Fig. 1. Rotation of angular coordinates (θ, φ) with reference to the origin T.

Let $I(i, j, k, t)$, $I^s(\theta, \varphi, t)$ and $I^s(\theta_k, \varphi_l, t)$ be, respectively, the input 3D $+ t$ intensity sequence, the intermediate list of scattered points and the final 2D$+t$ sequence. In order to compute the final sequence, we proceed as the following:

- First we convert the discrete 3D$+t$ sequence, considered from a selected point of view, to a set of scattered values of dimension 3. First couple of variables are spherical angles (θ, φ) deduced from cartesian ones by trigonometrical conversions, and third one is time t. θ and φ computed from discrete 3D do not correspond to grid-positions in the output sampled image. Therefore, we obtain an intermediate list of intensity values $I^s(\theta, \varphi, t)$.
- Secondly, we attempt to fit a smooth 2D $+ t$ surface through the non-uniform distribution of $I^s(\theta, \varphi, t)$ data samples. For this purpose, we manage to represent our surface as a sum of weighted and shifted synthesis functions. B-splines stand apart for a good compromise between quality (namely high fidelity reconstruction and regularity) and computational issues. A framework introduced in [7] is dedicated to recover such samples, in the 2D case. The implemented algorithm makes use of a coarse-to-fine hierarchy of control lattices in order to generate a sequence of bi-cubic B-spline functions whose sum approaches the desired interpolation function. Since, we aim at estimating a 2D$+t$ sequence, we carry out this method extended to a higher dimension.

B. Angular velocity estimation

The position of a moving pixel on the spheric surface is determined by the couple of angles (θ, φ), defined in figure 2 and the following equality : $\overrightarrow{OM} = R\overrightarrow{e_r}$.

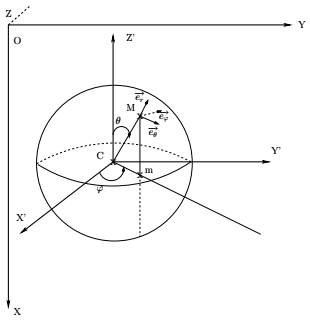


Fig. 2. Spherical system of coordinates.

The optical flow constancy constraint leads to:

$$I(R\overrightarrow{e_r}, t) \approx I(R\overrightarrow{e_r} + \partial\overrightarrow{Re_r}, t + \partial t), \tag{2}$$

where $R\overrightarrow{e_r} + \partial\overrightarrow{Re_r}$ is the displacement of the local image region at $(R\overrightarrow{e_r}, t)$ after time ∂t. $\partial\overrightarrow{e_r}$ is:

$$\partial\overrightarrow{e_r} = \partial\theta\partial\overrightarrow{e_\theta} + \sin(\theta)\partial\varphi\partial\overrightarrow{e_\varphi}$$

Hence, equation 2 becomes, for a unity sphere ($R = 1$):

$$I(\overrightarrow{e_r}, t) = I(\overrightarrow{e_r} + \partial\theta\partial\overrightarrow{e_\theta} + \sin(\theta)\partial\varphi\partial\overrightarrow{e_\varphi}, t + \partial t)$$

Expanding the left-hand side of this equation in a Taylor series, subtracting $I(\overrightarrow{e_r}, t)$ on both sides and dividing by ∂t yields:

$$\frac{\partial I}{\partial\theta}\frac{\partial\theta}{\partial t} + \sin(\theta)\frac{\partial I}{\partial\varphi}\frac{\partial\varphi}{\partial t} + \frac{\partial I}{\partial t} = 0,$$

Let $\dot{\theta}$ and $\dot{\varphi}$ be temporal derivative of respectively θ and φ, I_θ and I_φ spatial derivatives with reference to respectively θ and φ. The optical flow constraint on a sphere is:

$$I_\theta\dot{\theta} + \sin(\theta)I_\varphi\dot{\varphi} + I_t = 0 \tag{3}$$

C. Resolution

In order to estimate a dense motion field we use a variational formulation. We build a functional whose minimum, with reference to $\overrightarrow{v}$, corresponds to the equation 3 solution.

$$E_1(\overrightarrow{v}) = \int_\theta \int_\varphi |I_\theta \dot{\theta} + \sin(\theta) I_\varphi \dot{\varphi} + I_t|^2 d\theta d\varphi$$

Since equation 3 is under-determined, it is necessary to use an additional constraint to solve it. We then add a second term to our functional that penalizes high spatial deformations of $\overrightarrow{v}$:

$$E_2(\overrightarrow{v}) = \int_\theta \int_\varphi |\overrightarrow{\nabla}\dot{\theta}|^2 + |\overrightarrow{\nabla}\dot{\varphi}|^2 d\theta d\varphi$$

Finally, our functional E, is, with respect to $\overrightarrow{v}$:

$$E(\overrightarrow{v}) = E_1 + \alpha E_2,$$

where α is a tuning parameter between the two terms that weights the importance of the regularization term E_2. Differentiation of E, with reference to $\dot{\theta}$ and $\dot{\varphi}$ yields a set of Euler-Lagrange equations (reader is referred to [8]). Discretization of the latter with finite differences leads to a great linear system. We solve the minimization of the linear system within a Gauss-Seidel iterative scheme.

Moreover, we can estimate a piecewise continuous motion field by adding the extra assumption that the optical flow is locally constant. To find the flow vector at any point $M(\theta_M, \varphi_M)$, we minimize the function:

$$\sum_{m \in N(M)} W(m)(I_\theta(m)\dot{\theta_M} + \sin(\theta(m)) I_\varphi \dot{\varphi_M} + I_t(m))^2, \tag{4}$$

Where $W(m)$ is a normalized neighborhood weighting function and $N(M)$ is a neighborhood region surrounding M. The minimization problem is solved using singular value decomposition. As already outlined, both global and local resolution methods are similar to classical $2D$ ones, except that in our case, all assumptions are available in the spherical system of coordinates. Implementation has hence the merit to be rapid and stable, with few parameters to adjust.

IV. Visualization

We are interested in visualizing our computed optical flow field. Recall that in previous section, we performed a structural modification in our input 3D + t data set. We have a 2D+t sequence with different spatial resolutions and structure organization. Computed velocity vector field have the same spatial resolution than the texture scalar sequence 2D + t, as well as a coherent structure. Consequently, instead of carrying out an inverse transformation to convert our vector field to the original configuration, we select a suitable tool to visualize our spherical texture as well as the velocity vector in each couple of angular coordinates (θ, φ). Since the underlying enclosing surface has a spherical geometry, map projections are good candidates to project the 3D input surface into a planar cartographic reference. We opt for the *azimuthal equal area projection* presenting relevant visualization properties. We have then built a novel map-projection based tool for visualizing scalar and vectorial information lying on spherical surfaces, called **Mapvis**. It displays 3D scalar and vectorial data sets, see [9] for a whole description. We may sum up map-projection stages by the following scheme:

$$(\lambda, \varphi) \longmapsto (r, \theta) \longmapsto (x, y)$$

Geographical coordinates of each point of the input sphere are transformed into polar, then, cartesian ones in the projection plan. Since, our transformed 3D+t datasets are indexed with spherical coordinates (θ, φ), and conversion from spherical to geographical coordinates is trivial, displaying results with **Mapvis** becomes a natural simple task.

V. Results

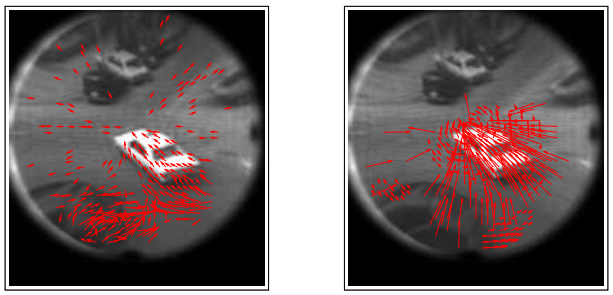
A. Biological context

In bio-cellular imaging, microscopic cell wall models are designed in order to analyze the behavior of some biological structures and find out their functional role. Due to some biological constraints, these models, called vesicles, have spheroid shapes. Biologists inject fluorescent markers on the spheroid surface, in order to segregate between objects interacting with the input light spot. These markers target at viewing some lipid exterior phases generally of weak thickness called *rafts*, moving on vesicle surfaces. Biologists are interested on analyzing behavior of rafts (shape and mainly movement, *etc.*) in order to find out their functional role (see [10] for details). We estimate *raft* motion on the spheroid surface by computing optical flow velocity field on the simulation temporal sequences.

B. Validation

In order to validate optical flow computation on a sphere, we first used the well known *urban taxi* sequence usually used to test 2D optical flow computation. In order to display results for 3D+t sequence, we map each frame of the input 2D + t series into 3D sphere with suitable parameters. We then compute optical flow by both local and global algorithms on the designed spheres and project both textures and velocity fields into 2D cartographic references. In figure 3, we visualize results of movement of the white taxi toward the right road, on the scene mapped onto a spherical support. The projection origin is set as the taxi center for texture as well as velocity field on the showed frame.

We notice, in this example, that global regularization generates a smoother motion field. Local method, tested on a 20 pixel neighborhood, give similar results on moving structures *i.e* white taxi and black car on the right bottom side of the hemisphere. However it also yields some scattered vectors mainly in polar zone, where deformation introduced by our cartographic projection are most important. In both cases, optical flow computation produce a set of correct vector fields (from amplitude and direction point of views), perfectly tangent to the enclosing spherical surface. Besides,



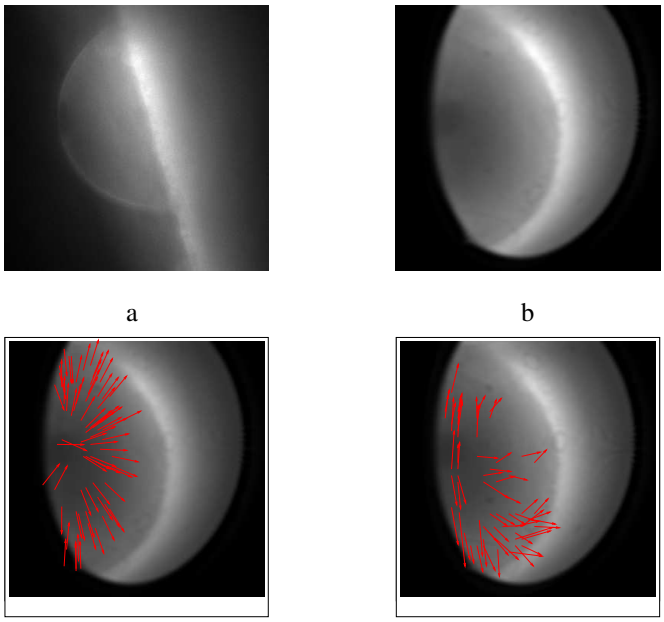
a b

Fig. 3. Visualization of taxi sequence texture and velocity field computed, from right to the left, by global then local optical flow method on a 20 pixel neighborhood

3D cartesian displacements can be estimated with trivial trigonometrical operations.

C. biological data

We tested both methods on several 3D+t cell wall simulation sequences. Due to the experimental context, images are blurred and have a very low contrast resolution, which doesn't ease optical flow computation. We display in figure 4, spherical optical flow computation results on a given sequence showing a *raft* moving on the vesicle surface, from the middle-left side to the right one on the front hemisphere. Quarter (a) shows a prospective projection of the vesicle holding the *raft* on the middle left border. In (b) we present a cartographic reference of the same frame by projecting the spheroid vesicle texture taking as origin the *raft* center. We can see that the *raft* shape is better recovered. In (c) we add to the cartographic reference the projected motion vector field computed by the global method and in (d) the projected one estimated by the local approach, in a 20 pixel neighborhood. Again, we can see that global method seem to give better results. Moreover, both methods capture correctly the *raft* movement with a minimum cost.



a b c d

Fig. 4. Visualization of spherical optical flow computation for a 3D+t cell wall simulation sequence.

VI. Conclusion

In this paper, we have presented a new approach for optical flow computation. The central idea is to take advantage from the geometry of the support on which structures are evolving to build an adapted model, involving a relevant data transformation. Since movement of structures of interest is embedded on a spherical surface, it has only two degrees of liberties. Consequently, we transform the $3D + t$ input temporal sequence to a $2D + t$ video sequence, indexed with angular coordinates. Local and global optical flow and smoothness regularization constraints are formulated on angular coordinates and they are solved with a suitable variational method. Compared to an extension of a classical variational model to a higher dimension, our model is more adapted to the processed data set, where the classical 3D optical constraint is not relevant, and obviously simpler and faster. Moreover, it is possible to visualize directly, the transformed sequence and the computed angular vector field with an adapted tool, with relevant visualization properties. The inverse data transformation is not necessary to highlight motion computation results. However, it is possible to deduce 3D cartesian displacements with trivial trigonometrical operations. In a future work, it is possible to improve the accuracy of optical flow computation, mainly where first order constancy hypothesis doesn't hold, by building hierarchical models taking also advantage from the 2D parametrization.

References

[1] B. Horn and Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[2] P. Pérez and F. Heitz, "Global bayesian estimation, constrained multiscale markov random fields and the analysis of visual motion," in *Int.Maxent Workshop*, no. 12, Paris, July 1992.

[3] J. Bergen, P. Anandan, J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *Proceedings of ECCV*, May 1992, pp. 237–248.

[4] J. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of Visual Communication and Image Representation*, vol. 6, no. 4, pp. 348–365, decembre 1995.

[5] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, Canada, 1981, pp. 674–697.

[6] I. Jonathan, *Datums and map projections for remote sensing, GIS, and surveying*. Scotland: Whittles Publishing: Caithness, 2000.

[7] S. Lee, G. Wolberg, and S. Shin, "Scattered data interpolation with multilevel b-splines," *IEEE Transactions on Visualization and Computer Graphics*, vol. 3, no. 3, pp. 228–244, July-September 1997.

[8] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, springer ed. New York: Series in computational physics, 1984.

[9] W. Rekik, D. Béréziat, and S. Dubuisson, "Mapvis: a map-projection based tool for visualizing scalar and vectorial information lying on spheroidal surfaces," in *Proceedings of international conference of information visualization, IV05*, London, the United Kingdom, July 2005.

[10] K. Simons and E. Ikonen, "Functional rafts in cell membranes," *Nature*, vol. 387, pp. 569–572, 1998.

A.3 Reconstruction 3D+t

[1] W. Rekik, D. Béréziat, and S. Dubuisson. Recovering 3D+t scenes for 3D and 2D+t data by motion compensation using a prior surface model. Technical report, Laboratoire d'Informatique de Paris 6, August 2010.

[2] W. Rekik, D. Béréziat, and S. Dubuisson. Fusion de données spatiales (3D) et temporelles (2D+t) pour la reconstruction de scènes 3D+t : deux approches. Technical report, Laboratoire d'Informatique de Paris 6, juin 2008.

[3] W. Rekik, D. Béréziat, and S. Dubuisson. 3D+t reconstruction in the context of locally spheric shaped data observation. In *International Conference on Computer Analysis of Images and Patterns (CAIP)*, pages 482–489, Vienna (Austria), August 2007.

Research report LIP6

Recovering 3D+t scenes from 3D and 2D+t data by motion compensation and using a prior surface model.

Wafa Rekik[1], Dominique Bérézíat[2,3], Séverine Dubuisson[2].

September 2010

[1]Present address: wafa.rekik@gmail.fr
[2]Department LIP6, Université Pierre et Marie Curie, 4 place Jussieu 75004, Paris, France.
[3]Corresponding author at: UPMC 4, place Jussieu 75004, Paris, France; tel: +33 1 44 27 47 71; fax: +33 1 44 27 53 53; mail: dominique.bereziat@upmc.fr

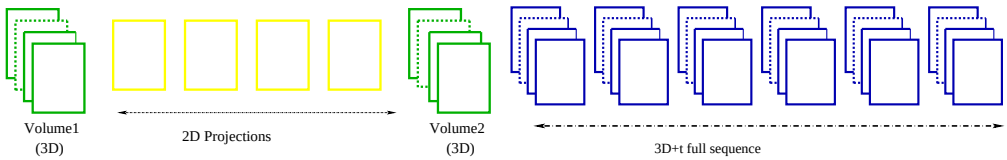


Figure 1: Recovering a 3D+t sequence from a couple of 3D images framing a series of 2D projections.

Abstract

In this paper, we address 3D+t shape recovery from 3D spatial data and 2D+t temporal sequences. This reconstruction is particularly challenging due to the great deal of in-depth information loss observed on the 2D+t temporal sequence. To handle this critical lack of information, we use a geometrical local constraint defined by a spherical topology arising in number of computer vision applications. Our approach consists in gradual 2D-to-3D restoration by motion compensation using a brightness constancy constraint and a spatial regularity criterion. Data prior geometrical model implies that structures of interest evolve on a spherical support. Motion has then only two degrees of liberty. We build then an adapted 2D model in the surface domain, where movement assumptions are more relevant, using a spherical parametrization. We restore 2D+t surfaces parametrized by spherical angular coordinates that we display using an adapted visualization tool.

Keywords

3D+t reconstuction, optical flow, motion compensation, spherical parametrisation.

1 Introduction

In some computer vision applications, we deal with volumetric, *i.e.* 3D, acquisitions framing temporal, *i.e.* 2D+t, ones (Fig. 1). First type of acquisitions provides a description of the 3D scene geometry, thus, purely spatial information. Second ones observes 2D object motion providing temporal and partial spatial information. A possible approach to mine exhaustively these complementary datasets is to carry out a complete spatio-temporal or a 3D+t scene reconstruction. 3D+t sequences are restored by recovering the 3D original volume from each 2D frame belonging to the 2D+t sequence.

As the matter of fact, 3D reconstruction is an inverse problem. In this particular case, it is also an ill-posed problem since a single 2D frame is hardly sufficient to recover the 3D original volume. Indeed, this frame exhibits a great deal of spatial distortions and in-depth loss of information caused by the projective transformation that reduces the 3D real structure into a 2D image. To handle this critical lack of information, we introduce a prior data model. It consists in a geometrical constraint defined on a spherical topology. Data observing locally spheric shaped objects can be issued from remote sensing acquisitions and especially in meteorology, from astronomical ones focusing on the Sun and in bio-cellular microscopic imagery. A relevant biologic application is described in [9] where biologists study dynamic of structures of interest evolving on a spherical

cell simulation surface. In this context, for instance, spatial datasets consist in multi-focus acquisitions. Hence, they are not actual 3D images since they are formed by a series of equal sized slices, where signal in each slice is bothered by luminosity coming from adjacent ones. A spatial deconvolution stage is therefore needed when biologists calibrate correctly the video microscopy system to cell wall simulation experiments. Due to this limitation and to respect generality, we use synthetic data to validate our algorithm.
Our approach uses this geometrical prior constraint in order to select a relevant reconstruction strategy. In the context of locally spheric shaped data, the information of interest is embedded on the outer curved surface. This allows the 3D-to-2D transformation to be modeled simply with a surface based reconstruction approach. On one hand, inverse 3D restoration is performed gradually by motion compensation involving a spatial regularity criterion that balances the in-depth loss of information. On the other hand, since movement has only two angular degrees of liberty, we opt for a spherical parametrisation to investigate motion in the surface domain instead of building a 3D model. For the latter, computation of spatial derivatives is incoherent and noisy, mainly with reference to the normal to the surface. Besides movement constraints are not relevant. We restore then spherical surfaces, parameterized by angular coordinates, that we display directly by an adapted visualisation tool, called MAPVIS [8].

2 State-of-the-art

In the literature, a wide scope of approaches addresses the 3D reconstruction problem. Volume based methods aim at restoring an accurate description of luminosity in each voxel of the 3D reconstructed image, while surface based ones provide a 3D model of the object of interest. This model requires shape and texture recovery of the outer visible side of the observed object. Namely, tomographic reconstruction [7] focuses on restoring the original volume from multiple projective views. Stereo-vision [4] approaches make use in general of the triangulation principle in order to recover the surface shape of selected targets. Moreover, 2D-to-3D registration methods reconstruct 3D features from only one projective view. This registration consists in identifying a geometrical transformation that aligns 2D frames with 3D datasets often issued from a different modality of acquisition. This alignment determines their relative position and orientation and then locates them in the coordinate system of the 3D images. It is of growing interest in the medical imaging field mainly to register 3D pre-operative images (like MRI or CT images) into 2D series of intra-operative frames acquired during a surgical intervention. Registration may operate on feature-based methods or intensity based ones. Reader is referred to [2, 6]. Tomography and stereovison, namely, require multiple projective views to achieve the reconstruction task. Therefore, approaches aligning 2D projection images to 3D datasets appear to be adapted to the context of our study. However these approaches fit into a registration framework rather than a reconstruction one, unless other projective views are available, which is quite unusual. They can be used alternatively for a preprocessing stage for some aforementioned applications. We then build an adapted reconstruction model involving a prior geometrical constraint.

3 Reconstruction by motion compensation

Our approach is based on the existence and the knowledge of a functional p that reduces every volumetric structure at each instant into a 2D frame. The central idea is to carry out the 3D+t reconstruction using a couple of 3D data acquired at two different instants as well as the 2D+t sequence acquired meanwhile. Since the intermediary video sequence presents an interesting temporal resolution, we propose to restore the underlying 3D structure of each of its frame. The latter provides only a partial description of the real 3D structure that we aim to estimate at a given instant t. We attempt to compensate this lack of information by the mean of a matching procedure with the 3D volumetric image estimated at $t-1$. For this purpose, we compute a displacement field matching both 3D structures. We introduce, consequently, a motion hypothesis expressing the assumption that a moving voxel keeps the same gray value over time. It is identical to the well known optical flow constraint [3]. It leads to a fundamental equation that we solve gradually in order to rebuild the whole 3D+t sequence, using the couple of 3D data as initial border conditions.

3.1 Modeling, assumptions

We carry out the 3D+t reconstruction using a series of 2D+t video sequence acquisitions intersected with 3D volumetric ones. We opt for the following notations for the remainder of the paper:

- $\mathbf{X}$ a 3D vector position, *i.e.* $\mathbf{X} = (x, y, z)$;
- $\mathbf{x}$ a 2D vector position, *i.e.* $\mathbf{x} = (x, y)$;
- $\mathbf{W}$ a 3D vector field, *i.e.* $\mathbf{W} = (u, v, w)$;
- $I(\mathbf{X}, t)$ a 3D+t sequence;
- $I_{2\text{D}}(\mathbf{x}, t)$ a 2D+t sequence;
- $\hat{I}(\mathbf{X}, t)$ the estimation of I at $(\mathbf{X}, t)$;

Concretely, we have two 3D structures at two different instants: $I^1 = I(\mathbf{X}, t_1)$ and $I^2 = I(\mathbf{X}, t_2)$. Between $t = t_1$ and $t = t_2$, we also dispose of a video sequence $I_{2\text{D}}(\mathbf{x}, t)$ describing all the 2D reductions of the $I(\mathbf{X}, t)$ structures evolving in this duration. Each 2D frame $I_{2\text{D}}$ is a projective sight, at a given instant t, of a 3D original volume. This projection only depends on the data acquisition mechanism. We model it by a linear and stationary transformation, called projection p, given by:

$$p(I)(x, y) = \int_{\mathbb{R}} I(x, y, z)h(z)dz, \tag{1}$$

where $h(z)$ is a function describing the interaction between observed objects and the input light signal. p integrates contributions of all 3D slices to produce a unique 2D image. It introduces then an important loss of information in the $z-$direction. In order to reconstruct gradually the 3D+t sequence, we estimate the 3D structure at a given instant t, using the data $I_{2\text{D}}(\mathbf{x}, t)$ extracted from the 2D+t sequence. We then minimize the discrepancy between $p(\hat{I})$ and $I_{2\text{D}}$. A straightforward data model equation is:

$$p(\hat{I})(\mathbf{x}, t) - I_{2\text{D}}(\mathbf{x}, t) = 0, \ \forall t \in [t_1, t_2] \tag{2}$$

Since objects of interest evolve slowly from t to $t-1$, we can match $\hat{I}(\mathbf{X}, t)$ with the 3D structure $I(\mathbf{X}, t-1)$ known at the previous instant. This matching is quantified in terms of a displacement

vector field $\mathbf{W}$ between both images. It expresses the assumption that a moving voxel keeps the same gray value over time yielding the following motion constraint, similar to optical flow one:

$$\hat{I}(\mathbf{X}, t) = I(\mathbf{X} + \mathbf{W}, t-1) \tag{3}$$

We estimate a retrogress vector field defining voxel displacement from $t-1$ to t for notation conveniences (we represent $-\mathbf{W}$ in Section 4). Expanding the left-hand side of Equation (3) in a first order Taylor series leads to:

$$\hat{I}(\mathbf{X}, t) = \nabla I(\mathbf{X}, t-1).\mathbf{W} + I(\mathbf{X}, t-1), \tag{4}$$

where $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})$ is the gradient operator. Since p is a linear operator, injection of the Equation (4) in the earlier data model constraint (2) provides the 2D data motion constraint related to $\mathbf{W}$:

$$p(\nabla I(\mathbf{X}, t-1).\mathbf{W}) + p(I(\mathbf{X}, t-1)) - I_{2\text{D}} = 0 \tag{5}$$

We compute the velocity vector field $\mathbf{W}$ at each instant t.

3.2 Formulation in the spherical context

Let us point out that, in the context of spherical shaped data observations, structures of interest evolve on the surface of a ball of constant ray. Even-though the support is embedded on a 3D-space, movement has only two degrees of liberty. It is then relevant to estimate the vector field using a spherical surface parametrisation. Let's establish equations formulated in the previous section in the spherical context. For this purpose, we introduce two 2D images I^s and $I^s_{2\text{D}}$ defined by the following substitutions (see Fig. 2):

$$I^s(\theta, \varphi, t) = I(x(\theta, \varphi), y(\theta, \varphi), z(\theta, \varphi), t) \tag{6}$$

$$I^s_{2\text{D}}(\theta, \varphi, t) = I_{2\text{D}}(x(\theta, \varphi), y(\theta, \varphi), t) \tag{7}$$

with the following parametrisation:

$$\begin{aligned} x(\theta, \varphi) &= R\cos(\varphi)\sin(\theta) \\ y(\theta, \varphi) &= R\sin(\varphi)\sin(\theta) \\ z(\theta, \varphi) &= R\cos(\theta) \end{aligned}$$

The spherical coordinates φ and θ respectively stand for the longitude and co-latitude angles. Equation (6) expresses merely passage from cartesian coordinates to spherical ones. Equation (7) involves the mapping of the 2D frame texture into a hemisphere with same diameter that the 3D known balls. Data model constraint (2) becomes:

$$p(\hat{I}^s)(\theta, \varphi, t) - I^s_{2\text{D}}(\theta, \varphi, t) = 0 \tag{8}$$

Let's now set up p in the sphere reference mark schemed in Fig. 2. Operator p integrates, in a plane including both geographical poles, contributions of frontal and dorsal hemispheres.
Assuming that frontal and dorsal interactions with the input light signal are the same, p is defined by:

$$p(I^s)(\theta, \varphi) = I^s(\theta, \varphi) + I^s(\theta, \pi - \varphi) \tag{9}$$

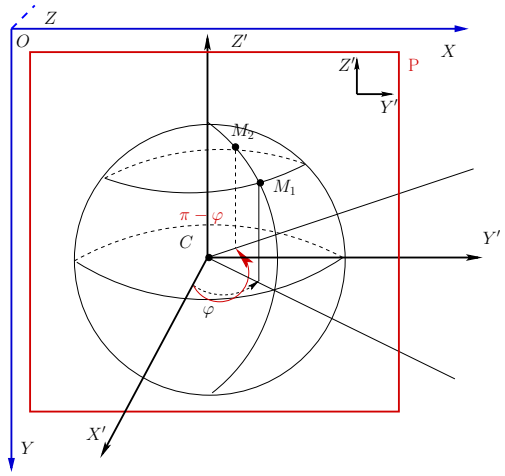


Figure 2: System of spherical coordinates. The projection plane corresponding to p is $P = (C, Y', Z')$.

Matching between $\hat{I}^s(.,t)$ and $I^s(.,t-1)$ are now expressed as performed in the cartesian parametrisation by Equation (3). According to Fig. 2 notations, the position of a moving point on the spherical surface is determined by the couple of angles (θ, φ) and the following equality : $\overrightarrow{CM} = R\overrightarrow{e_r}$, where R is the sphere radius and $(\overrightarrow{e_r}, \overrightarrow{e_\theta}, \overrightarrow{e_\varphi})$ the Frenet local coordinate system related to M. Thus:

$$\hat{I}^s(R\overrightarrow{e_r}, t) = I^s(R\overrightarrow{e_r} + \partial R\overrightarrow{e_r}, t-1), \tag{10}$$

where $\overrightarrow{e_r} + \partial R\overrightarrow{e_r}$ is the elementary displacement at $R\overrightarrow{e_r}$ after the duration ∂t. Since R is constant, $\partial\overrightarrow{e_r}$ is defined by :

$$\partial\overrightarrow{e_r} = \partial\theta\partial\overrightarrow{e_\theta} + \sin(\theta)\partial\varphi\partial\overrightarrow{e_\varphi} \tag{11}$$

Inserting right-hand side of Equation (11) in the equality (10) leads to, for a unit sphere ($R = 1$):

$$\hat{I}^s(\overrightarrow{e_r}, t) = I^s(\overrightarrow{e_r} + \partial\theta\partial\overrightarrow{e_\theta} + \sin(\theta)\partial\varphi\partial\overrightarrow{e_\varphi}, t-1) \tag{12}$$

The right member of Equation (12) is linearized using a first order Taylor expansion. It yields the following equation:

$$\begin{aligned}\hat{I}^s(\theta,\varphi,t) = I^s_{t-1}(\theta,\varphi,t-1) + \frac{\partial I^s(\theta,\varphi,t-1)}{\partial\theta}\frac{\partial\theta}{\partial t} \\ + \sin(\theta)\frac{\partial I^s(\theta,\varphi,t-1)}{\partial\varphi}\frac{\partial\varphi}{\partial t}\end{aligned} \tag{13}$$

Inserting second member of Equation (13) in (8) and using linearity of the transformation p leads to:

$$\begin{aligned} p\Big(\frac{\partial I^s(.,t-1)}{\partial \theta}\frac{\partial \theta}{\partial t} + \sin(\theta)\frac{\partial I^s(.,t-1)}{\partial \varphi}\frac{\partial \varphi}{\partial t}\Big)(\theta,\varphi) \\ + \underbrace{p(I^s)(\theta,\varphi,t-1) - I^s_{2\mathrm{D}}(\theta,\varphi,t)}_{I'^s_{2\mathrm{D}}} = 0 \end{aligned} \tag{14}$$

Finally, the data model constraint in the spherical context is:

$$p\Big(\frac{\partial I^s(.,t-1)}{\partial \theta}\frac{\partial \theta}{\partial t} + \sin(\theta)\frac{\partial I^s(.,t-1)}{\partial \varphi}\frac{\partial \varphi}{\partial t}\Big) + I'^s_{2\mathrm{D}} = 0 \tag{15}$$

3.3 Numerical Resolution

We compute an angular velocity field defined by the couple of angular variations $(\dot{\theta}, \dot{\varphi})$:

$$\left\{ \begin{array}{rcl} \dot{\theta} & = & \frac{\partial \theta}{\partial t} \\ \dot{\varphi} & = & \frac{\partial \varphi}{\partial t} \end{array} \right. \tag{16}$$

For this sake, we have to solve the motion constraint given in (15). Let us introduce the aforementioned p expression in Equation (15):

$$\begin{aligned} p\Big(\frac{\partial I^s(.,t-1)}{\partial \theta}\frac{\partial \theta}{\partial t} + \sin(\theta)\frac{\partial I^s(.,t-1)}{\partial \varphi}\frac{\partial \varphi}{\partial t}\Big) = \\ \frac{\partial I^s(\theta,\varphi,t)}{\partial \theta}\dot{\theta}(\theta,\varphi) + \frac{\partial I^s(\theta,\pi-\varphi,t-1)}{\partial \theta}\dot{\theta}(\theta,\pi-\varphi) + \\ \sin(\theta)\Big(\frac{\partial I^s(\theta,\varphi,t-1)}{\partial \varphi}\dot{\varphi}(\theta,\varphi) + \\ \frac{\partial I^s(\theta,\pi-\varphi,t-1)}{\partial \varphi}\dot{\varphi}(\theta,\pi-\varphi)\Big) \end{aligned} \tag{17}$$

where $(\dot{\theta}(\theta,\varphi), \dot{\varphi}(\theta,\varphi))$ is the couple of angular velocity fields on the frontal hemisphere and $(\dot{\theta}(\theta,\pi-\varphi), \dot{\varphi}(\theta,\pi-\varphi))$ the couple of fields on the dorsal one. Equation (15) involves four velocity fields. This is due to the transformation p that introduces a positioning ambiguity. Indeed, it is not possible to undoubtedly state if the original position of a given point on the projection plane is in the frontal or the dorsal hemisphere. In order to reduce variable space, we suppose that these velocity fields are related. Let's express relation between velocity fields on the frontal as well as on the dorsal hemisphere. This relation takes into account possible object occlusions. In fact, objects with same latitude and a double-quadrant adjusted longitude are mapped at exactly the same position in the projection plane (see Fig. 3). To circumvent this ambiguity of positioning, we assume that:

$$\left\{ \begin{array}{rcl} \dot{\theta}(\theta,\pi-\varphi) & = & \dot{\theta}(\theta,\varphi) \\ \dot{\varphi}(\theta,\pi-\varphi) & = & -\dot{\varphi}(\theta,\varphi) \end{array} \right. \tag{18}$$

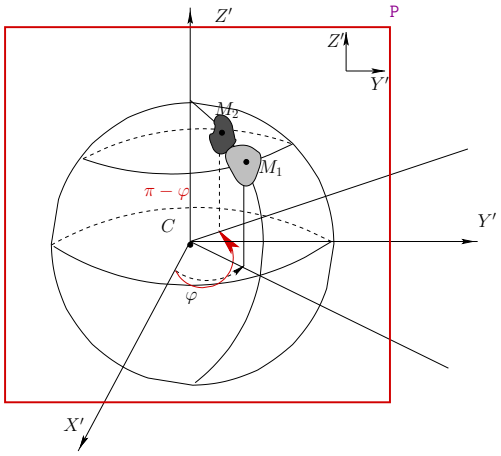


Figure 3: Occlusion of objects centered respectively in M_1 and M_2.

Under this assumption, number of variables is reduced to two. We estimate then velocities on the upper hemisphere, where θ has a range from 0 to π and φ from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$. For the remainder of the globe, displacement vectors are issued from the system of equations (18). Data motion constraint is merely respect to $\mathbf{v} = (\dot{\theta}, \dot{\varphi})$:

$$\begin{aligned} \mathbb{M}(\mathbf{v})(\dot{\theta},\dot{\varphi},t) = p(\frac{\partial I^s(.,t-1)}{\partial \theta})\dot{\theta} + \\ \sin(\theta)\big(\frac{\partial I^s(.,t-1)}{\partial \varphi}(\theta,\varphi) - \frac{\partial I^s(.,t-1)}{\partial \varphi}(\theta,\pi-\varphi)\big)\dot{\varphi} \\ + {I'}^s_{\text{2D}} = 0 \end{aligned} \tag{19}$$

Since this equation involves only two variables, we can build a bi-dimensional model to accurately recover motion. In order to estimate a dense motion field we use a variational formulation. We build a functional whose minimum, with respect to $\mathbf{v}$, corresponds to the solution of Equation (19):

$$E_1(\mathbf{v}) = \int_0^{\pi} \left(\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \mathbb{M}(\mathbf{v})^2 d\varphi \right) d\theta$$

Since Equation (19) is under-determined, it is necessary to use an additional constraint to solve it. An obvious way is to constrain the spatial variations of $\mathbf{v}$ using a L_2 norm [11]. This regularization penalizes high spatial deformations of $\mathbf{v}$ and implicitly important variations with reference to the data known at a previous time:

$$E_2(\mathbf{v}) = \int_0^{\pi} \left(\int_{\frac{-\pi}{2}}^{\frac{\pi}{2}} \left(\|\nabla\dot{\theta}\|^2 + \|\nabla\dot{\varphi}\|^2 \right) d\varphi \right) d\theta$$

Other types of regularization may be used (L_1 norm, image driven norm, ...). Finally, our functional E, is, with respect to $\mathbf{v}$: $E = E_1 + \alpha E_2$, where α is a tuning parameter between the two

terms that weights the importance of the regularization term E_2. Differentiation of E, with reference to $\dot{\theta}$ and $\dot{\varphi}$ yields a set of Euler-Lagrange equations [1]. Discretization of the latter with finite differences leads to a linear system, that we solve using a Gauss-Seidel iterative scheme.
Our goal, as schemed in Fig. 1, is to recover a full 3D+t sequence from the 2D+t series of data and the couple of 3D samples framing the temporal interval. The central idea is to estimate gradually velocity field between each 2D frame extracted from the 2D+t at a given instant t and the 3D structure previously estimated at $t-1$, as detailed in Section 3.1. Under the constant brightness assumption from time $t-1$ to t, an estimation of $\hat{I}^s(.,t)$ is recovered from the data $I^s(.,t-1)$ and angular variations $(\dot{\theta},\dot{\varphi})$ by the following:

$$\hat{I}^s(\theta,\varphi) = I^s(\theta+\dot{\theta},\varphi+\dot{\varphi},t-1) \tag{20}$$

However angular coordinates $\theta+\dot{\theta}$ and $\varphi+\dot{\varphi}$, computed from discrete 2D couples, do not generally correspond to grid-positions in the output sampled image sequence I^s: they yield a set of scattered data. We attempt, therefore, to fit a smooth surface through the non-uniform distribution of these data samples. For this purpose, we represent our surface as a sum of weighted and shifted synthesis functions. B-splines stand apart for a good compromise between quality (namely high fidelity reconstruction and regularity) and computational issues, reader is referred to [10]. We then use the method introduced in [5] and dedicated to recover such samples. The implemented algorithm makes use of a coarse-to-fine hierarchy of control lattices in order to generate a sequence of bi-cubic B-spline functions whose sum approaches the desired interpolation function.
In order to restore gradually the complete 3D+t sequence, we adopt a quite simple reconstruction strategy. We use, respectively, a forward progression procedure taking as first 3D available image, I^1 and a backward one using the final 3D available image I^2. Consequently, we generate two displacement vector fields at the median moment of the temporal interval $[t_1,t_2]$. The final 3D image is computed merely using the average of both fields. Let us point out that we restore surfaces parametrized by angular coordinates (θ,φ). To display results, we use an adapted visualisation tool called, MAPVIS [8], instead of carrying out inverse transformations in cartesian 3D+t grids. This tool displays complex information (scalar and vectorial) lying on real spheroid surfaces or projected ones. In the 3D case, MAPVIS projects data embedded on the actual globe observed part. Moreover, it improves visualisation of texture in perspective sight views of spheric shaped structures. This tool is based on a suitable planar map projection that unrolls the curved surface around a given origin. Therefore, varying the projection origin around the surface allows to observe different views of the sphere. Since the selected map projection minimizes distortions around the projection origin, the closer this point the more accurate the data recovery. Equatorial aspect of map projection, with reference to an origin lying on the equator displays the frontal hemisphere of the observed globe and polar one, with reference to the north geographical pole displays information lying on the northern hemisphere.

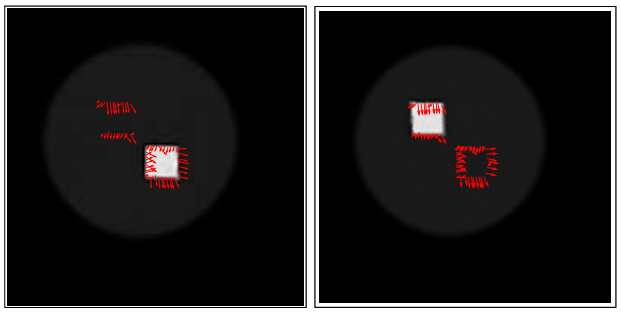
4 Results

We present, in this Section, a series of experiments in order to assess the performance of the proposed approach. They consist in computing the displacement vector field matching a 2D frame with an anterior 3D structure. As real data are not available, we have designed two sets of simple synthetic data. First one is composed of a couple of spheres $I(.,t_1), I(.,t_2)$ holding two squares

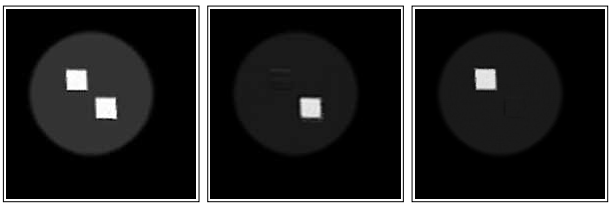
lying on respectively the frontal and dorsal hemispheres. From t_1 to t_2, the frontal square center moves towards the right-bottom direction and the dorsal square evolves merely downwards. We simulate the 3D-to-2D transformation of $I(.,t_2)$ yielding $I_{2D}(.,t_2)$ and we compute the spherical representation of I and I_{2D} using Equations (6,7). We then determine angular variation fields $(\dot{\theta},\dot{\varphi})$ matching data $I^s(.,t_1)$ with the projection $I^s_{2D}(.,t_2)$ in order to estimate $\hat{I}^s(.,t_2)$ by motion compensation. Second set of data is quite identical to the first one, except that squares occlude each other in the 2D projection. Results are displayed in cartesian coordinates using MAPVIS. We display in Fig. 4 and 5 motion estimation results for respectively first and second set of synthetic data. Let us point out that motion computation produces a set of correct vector fields (from amplitude and direction point of views), perfectly tangent to the enclosing spherical surface. Moreover, our reconstruction method handles ambiguities, due to the 2D data position, by determining which hemisphere it is evolving in. It is noticeable by visualisation of the difference between the estimation of 3D image and the original one, *i.e.* $I(.,t_2) - \hat{I}(.,t_2)$. Performance of our algorithm is not bothered by the more complex case showing an object occlusion (see Fig. 5) which does not ease the aforementioned structure distinction problem. However image difference shows also reconstruction errors, doubtless amplified by the double interpolation procedure involved, first by the computation of a smooth surface from samples recovered by motion compensation, second by the visualization of projective views [8]. These errors are localized in the border of moving objects. They may be caused by spatial derivative computations. Besides, they are due to the smoothing of motion discontinuities introduced by the spatial regularization of the estimated displacement vector field.

5 Conclusion

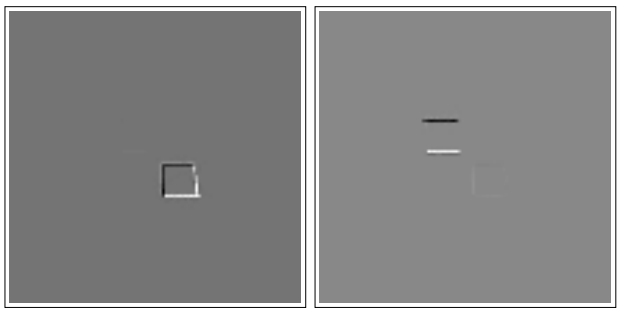
In this paper, we have presented an original approach dedicated to 3D+t scene reconstruction using 3D temporal data and 2D+t temporal sequences. It is based on motion compensation and involves a prior surface model. The latter concerns spherical topology arising in number of computer vision applications. In this context, structures of interest evolve on a spherical support. Although movement is embedded on a 3D domain, it has only two degrees of liberty. Consequently, it is more interesting to investigate motion in the surface domain than to build a 3D model. Consequently, we build a 2D model using a spherical parametrisation in order to compute angular velocity fields. Three dimensional spherical shapes are gradually restored by estimating angular variations. Matching is carried out under the assumption that a moving voxel keeps the same brightness over time. This assumption corresponds to the optical flow constancy constraint. Our approach recovers reliably the 3D structures and handles possible object occlusions. Thanks to the 2D modeling, implemented algorithms are not prohibitive compared to a full 3D approach. Moreover, we use an adapted tool, with relevant visualisation properties, in order to quickly display the restored surfaces parameterized by spherical angular coordinates. Currently, the whole sequence is computed frame by frame, causing errors cumulation along the sequence. This limitation could be circumvented using a spatio-temporal smoothness constraint computed globally on the whole sequence [12]. Discussing other reconstruction issues, it is possible to generalize the prior geometrical constraint to any regular surface. We propose also in another framework to solve the 3D+t scene recovery taken lower assumptions related to the data model. Moreover, we plan to apply our approach based on spherical topology to microscopic cell wall simulation acquisitions



(a) From left to right: visualisation of the frontal then dorsal hemisphere of the set $(I(.,t_1), \mathbf{W})$.

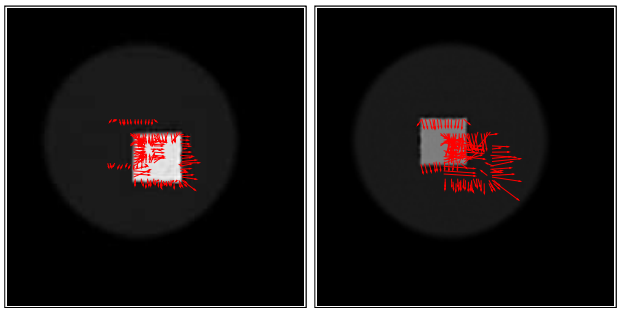


(b) From left to right: frontal view of projection $I_{2D}(.,t_2)$, frontal then dorsal hemisphere view of the estimated image $\hat{I}(.,t_2)$.

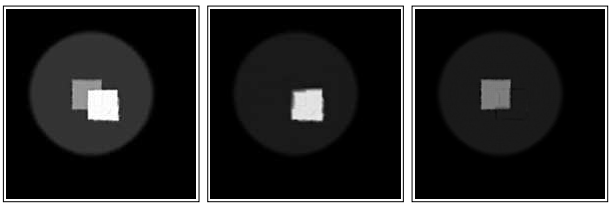


(c) From left to right: frontal then dorsal hemisphere view of the difference between original image and estimated one $I(.,t_2) - \hat{I}(.,t_2)$.

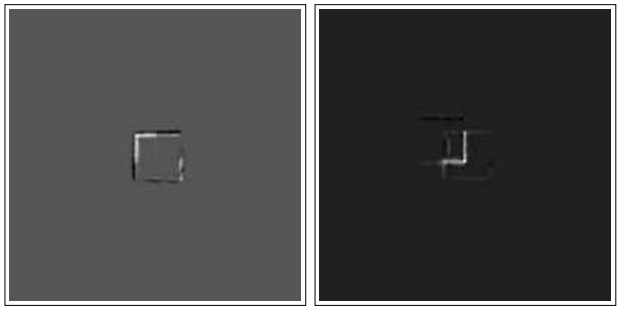
Figure 4: Result on sequence without occluding squares.



(a) From left to right: visualisation of the frontal then dorsal hemisphere of the set $I^s(.,t_1), \mathbf{W}$.



(b) From left to right: frontal view of projection $I_{2D}(.,t_2)$, frontal then dorsal hemisphere view of the estimated image $\hat{I}(.,t_2)$.



(c) From left to right: frontal then dorsal hemisphere of the difference between original image and estimated one $I(.,t_2) - \hat{I}(.,t_2)$.

Figure 5: Result on sequence with occluding squares.

in the context described in [9]. That requires a spatial deconvolution framework to reconstruct each sphere from the original multi-focus image.

References

[1] R. Glowinski. *Numerical Methods for Nonlinear Variational Problems*. Series in computational physics. Springer, New York, 1984.

[2] A. Gueziec, P. Kazanzides, B. Williamson, and RH. Taylor. Anatomy-based registration of CT-scan and intraoperative X-ray images for guiding a surgical robot. *IEEE Trans Med Imaging*, 17:715–728, 1998.

[3] B.K.P. Horn and B.G. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[4] L. Huei-Yung. *Computer vision techniques for complete 3D model reconstruction*. PhD thesis, University of New York, 2002.

[5] S. Lee, G. Wolberg, and S. Shin. Scattered data interpolation with multilevel B-splines. *IEEE Transactions on Visualization and Computer Graphics*, 3(3):228–244, July-September 1997.

[6] L. Lemieux, R. Jagoe, DR. Fish, ND. Kitchen, and DGt. Thomas. a patient-tocomputed-tomography image registration method based on digitally reconstructed radiographs. *Med Phys*, 21:1749–1760, 1994.

[7] F. Natterer. *The mathematics of computerized tomography*. Wiley, New York, 1986.

[8] W. Rekik, D. Béréziat, and S. Dubuisson. MAPVIS: a map-projection based tool for visualizing scalar and vectorial information lying on spheroidal surfaces. In *Proc. of IV05*, London, UK, July 2005.

[9] G. Staneva, M. Angelova, and K. Koumanov. Phospholipase A2 promotes raft budding and fission from giant liposomes. *Chem Phys Lipids*, 129:53–62, 2004.

[10] P. Thévenaz, T. Blu, and M. Unser. Image interpolation and resampling. In *Handbook of Medical Imaging*, volume 2, chapter 25, pages 393–420. Academic Press, 2000.

[11] A. Tikhonov. Regularization of incorrectly posed problems. *Sov. Math. Dokl.*, 4:1624–1627, 1963.

[12] J. Weickert and C. Schnörr. Variational optic flow computation with a spatio-temporal smoothness constraint. *Journal of Mathematical Imaging and Vision*, 14(3):245–255, May 2001.

Rapport de recherche LIP6

Fusion de données spatiales (3D) et temporelles (2D+t) pour la reconstruction de scènes 3D+t : deux approches.

Merging 3D Spatial Data and 2D+t Temporal Sequences for 3D+t Scene Reconstruction : Two Approaches.

Wafa Rekik[1], Dominique Béréziat[1,2], Séverine Dubuisson[1].

[1] Université Pierre et Marie Curie
[2] INRIA
Correspondance : Dominique.Bereziat@inria.fr

Mai 2008

Résumé

L'objet de cette étude est la reconstruction de scènes Temps/Volume, ou, 3D+t à partir de données statiques 3D encadrant des séquences temporelles 2D+t. La combinaison de ces données complémentaires fournit une séquence décrivant la dynamique 3D des structures d'intérêt en reconstruisant le volume original à partir de chaque image 2D de la séquence temporelle. Le problème de reconstruction inverse 2D-3D est complexe et mal posé à cause du manque d'information et de la distorsion spatiale induite par la vue projective dans chaque image 2D. Afin de compenser ce manque d'information critique, nous modélisons le processus de formation des images 2D à partir du volume 3D original et nous mettons en correspondance chaque image 2D avec le volume 3D connu à l'instant précédent. Cette mise en correspondance est établie/schématisée par un champ de vecteurs de déplacements quantifiant les variations de position de chaque voxel entre deux instants successifs. Pour ce faire, nous utilisons l'hypothèse de conservation de la luminosité. Nous proposons deux approches pour estimer le mouvement des voxels et nous reconstruisons la scène 3D+t de proche en proche estimé par compensation du mouvement. Des applications potentielles de cette étude concernent la biologie microscopique cellulaire et l'imagerie médicale.

Mots Clef

Reconstruction 3D, compensation du mouvement, contrainte du flot optique.

Abstract

The main focus of this paper is 3D+t shape recovery from 3D spatial data framing 2D+t temporal sequences. Merging these two kind of data yields a temporal sequence describing 3D dynamic of structures of interest by reconstructing the original volume from each 2D frame. 2D to 3D inverse reconstruction is a complex and ill posed problem due to the great deal of distortion and in-depth information loss generated by the projective view in each 2D frame. To handle this critical lack of information, we model relevantly the 3D-to-2D transformation that reduces each 3D image into a 2D one and we match each frame with the 3D image known at a previous time. Assuming that structures evolve scarcely between a couple successive moments, these images are matched by a displacement vector field measuring voxel position variation meanwhile. We embed the luminosty conservation constraint and we propose two approaches to estimate voxel movement

over time. We reconstruct gradually the 3D+t sequence by motion compensation. Some applications observing in different modalities 2D+t and 3D data can benefit from this study such as microscopical cell biology or medical imaging.

Keywords

3D reconstruction, motion compensation, optical flow constraint.

1 Introduction

La reconstruction d'une scène tridimensionnelle à partir d'indices visuels bidimensionnels ou d'images est l'une des problématiques les plus étudiées de nos jours. Elle trouve un intérêt en vision par ordinateur, en imagerie médicale et biologique ou dans des applications plus récentes comme le contrôle de qualité en industrie, *etc*. En particulier, nous nous intéressons au cas générique suivant : nous disposons d'images volumiques (3D) et de séquences d'images 2D (2D+t).

1. Les données **3D** définissent la géométrie en trois dimensions de la scène observée. Elles comportent des informations purement spatiales.
2. Les séquences **2D+t** sont porteuses d'informations temporelles et partiellement spatiales puisqu'elles décrivent en deux dimensions la dynamique des objets en mouvement.

En général, les données 3D sont contiguës aux séquences temporelles 2D+t (voir la figure 1). Pour exploiter d'une manière exhaustive l'information contenue dans les données 2D+t, il est intéressant de reconstruire une scène temps/volume ou 3D+t. Cette reconstruction est réalisée en restaurant une structure 3D originale à partir de chaque image 2D de la séquence temporelle intermédiaire. Les données 3D dont nous disposons peuvent alors servir de conditions initiales aux procédés de reconstruction 3D.

La situation décrite dans la figure 1 existe en imagerie biologique où le dispositif d'acquisition est en mesure de fournir une image 3D – en faisant varier la mise au point de ce dispositif et en utilisant un procédé de déconvolution calibrée – ou une séquence d'images mais focalisée à un plan précis dans la scène. On a donc un compromis entre précision temporelle et précision spatiale. Il existe un second cas de figure, en imagerie médicale. En acquisition par rayons X, on peut obtenir des images 3D par un scanner mais au prix d'une exposition plus longue et moins ciblée du patient. Mais on peut également se contenter d'une simple acquisition X projective, comme un angiogramme, qui permet d'obtenir la précision temporelle souhaitée. Le compromis se situe plutôt en terme de minimisation d'irradiation du patient.

Malheureusement, nous ne disposons pas, pour cette étude, de données réelles. Nous nous plaçons donc dans un contexte méthodologique dans lequel les données utilisées sont synthétiques ce qui a pour avantage de quantifier la pertinence des méthodes.

Dans cet article, nous proposons une extension à nos travaux de reconstructions 3D+t [13] qui fait maintenant abstraction de la forme des objets observés.

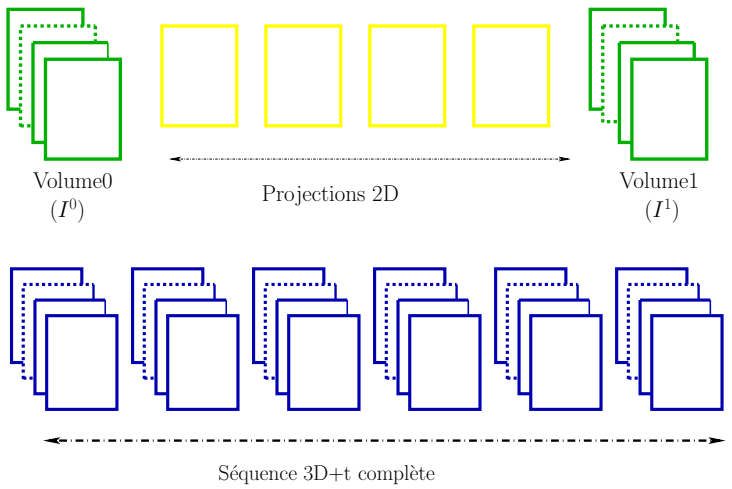


FIG. 1 – Reconstruction 3D+t complète à partir de volumes 3D ponctuels et d'acquisitions temporelles 2D+t.

Nous posons toutefois deux hypothèses. La première hypothèse concerne le procédé d'acquisition, en particulier, nous supposons connu le procédé qui transforme une scène 3D en une image 2D. Cette transformation est modélisée comme une transformée linéaire et stationnaire. La seconde hypothèse concerne la dynamique d'évolution des images 3D à reconstruire : dans cette séquence nous supposons que tous les points suivent une trajectoire de même valeur de niveaux de gris, autrement dit, nous utilisons une hypothèse de flot optique. L'hypothèse de transport de la luminosité doit se retrouver dans la séquence 2D+t et c'est ce qui nous permet de reconstruire la séquence 3D+t. Évidemment, l'utilisation d'une contrainte de type flot optique est un problème mal posé. Aussi nous proposons, à des fins de comparaison, deux méthodes de résolution. La première méthode est variationnelle : elle minimise une énergie qui mesure la discrépance du flot optique dans le domaine 2D en utilisant la transformée 3D-2D et la régularité spatiale du champ de déplacements. La second méthode élimine le critère de régularité spatiale du flot optique en calculant un champ constant par morceaux : le flot optique est alors calculé par moindres carrés à partir d'un système linéaire surcontraint.

2 Travaux précédents

Un grand nombre de méthodes, qui fournissent un modèle ou une image 3D à partir de vues projectives 2D, ont été proposées ces dernières années. Ces méthodes dépendent complètement du contexte d'étude : système d'acquisition et modalité d'image, type de traitement d'image souhaitée, nature de la scène observée, *etc*. Dans cette brève revue méthodologique, nous distinguons deux grandes classes de méthode.

La première classe regroupe toutes les approches proposant une reconstruction 3D effective. Cette reconstruction peut être complète en fournissant une image volumique complète : la scène 3D est décrite par une collection dense de voxels. Dans ce type de reconstruction, on privilégie l'information texturelle au détriment de la localisation des structures. La reconstruction peut aussi être incomplète en fournissant, par exemple, des reconstructions 3D de surfaces, généralement des objets ciblés dans la scène 3D. Dans ce type de reconstruction, on met l'accent sur l'information topologique ou la visualisation. Ces reconstructions sont obtenues à partir d'une série d'acquisition 2D. Nous pouvons alors identifier deux sous-groupes de méthodes selon que les acquisitions sont réalisées à partir d'un même angle de vue ou d'angles de vue différents. Dans le premier cas, observant la scène toujours d'un même point de vue, on peut focaliser l'acquisition à un plan de profondeur précis. Selon le type de capteur (focal, confocal) la focalisation est plus ou moins précise mais elle passe toujours par une étape de déconvolution [3] qui permet d'éliminer l'information parasite qui provient de plans de focalisation adjacents. Cette étape de traitement est difficile, elle nécessite des conditions d'acquisition calibrées qui ne sont pas toujours possibles. La calibration permet de rendre réalisable la déconvolution qui est un problème, rappelons-le, mal posé. Dans le second cas, où l'on observe donc la scène à différents points de vue, la situation est plus favorable. À partir d'un grand nombre d'acquisitions, on peut utiliser des techniques de reconstruction tomographique [11] pour obtenir une image 3D complète. Le principe de la tomographie repose sur l'inversion de la transformée de radon, qui est un problème bien posé. Cette technique est largement utilisée en imagerie médicale (CT-X, IRM). Avec deux ou trois caméras, on peut aussi reconstruire une image 3D par stéréoscopie [7]. Le procédé de reconstruction est purement géométrique et bien posé. Toutefois, la stéréoscopie nécessite un bon alignement des images, donc une phase de recalage qui peut être plus ou moins difficile selon la nature des images et des scènes. La stéréovision est utilisée en vision robotique mais aussi en imagerie médicale (images ultrasons 3D).

La seconde classe regroupe les méthodes de mise en correspondance 2D-3D.

Elles ont pour objectif de reconstruire des primitives 3D à partir d'une seule vue projective. Son rôle est de plus en plus important en imagerie médicale où elle est connue sous le terme de recalage 2D-3D [10]. Ce recalage est défini par la recherche de la meilleure transformation permettant de replacer un ensemble de données multimodales dans un espace tridimensionnel commun. En utilisant des images issues de diverses modalités, il permet de faire apparaître l'information qui manque à une modalité dans l'autre modalité. La majorité des applications pour ce genre de recalage se trouve en radiologie. Elles concernent l'alignement d'images scanner (CT) à une ou plusieurs images de projection à rayons X (images de simulateurs, images portiques, images générées par la fluoroscopie, *etc*). Ces méthodes d'alignement dépendent bien-sûr du contexte applicatif et des modalités utilisées. Nous pouvons distinguer essentiellement les méthodes basées sur des primitives caractéristiques comme notamment les contours [1] et les méthodes basées directement sur l'intensité de l'image [5, 8, 14].

Pour notre part, nous avons proposé une méthode de reconstruction 2D+t vers 3D+t utilisant des informations *a priori* sur la structure topologique des objets observés [13]. Cette méthode fournit d'abord une reconstruction de données géométriquement correctes mais souffrant de l'ambiguïté de positionnement dans la direction z, *i.e.* en profondeur. Cette ambiguïté est ensuite levée par une régularisation spatiale adéquate. C'est en continuité à ces travaux antérieurs que se situe notre étude. Notamment, les méthodes proposées ici permettent de s'affranchir des hypothèses faites sur la nature topologique des objets observés dans les acquisitions. Nous nous sommes placés dans un contexte générique où les acquisitions sont réalisées selon un point de vue unique, comme c'est le cas en imagerie biologique. Nous supposons connue la transformation de la scène 3D en une image 2D.

3 Modélisation et hypothèses utilisées

Nous menons la reconstruction 3D+t à partir d'acquisitions temporelles 2D+t, entrecoupées par des acquisitions volumiques 3D. Nous adoptons dorénavant les notations suivantes :

- $\mathbf{X}$ un vecteur position 3D, *i.e.* $\mathbf{X} = (x, y, z)$;
- $\mathbf{x}$ un vecteur position 2D, *i.e.* $\mathbf{x} = (x, y)$;
- $\mathbf{W}$ champ de vecteurs vitesse 3D, *i.e.* $\mathbf{W} = (u, v, w)$;
- $I(\mathbf{X}, t)$ une image 3D+t ;
- $I_{\text{2D}}(\mathbf{x}, t)$ une image 2D+t ;

– $\hat{I}(\mathbf{X}, t)$ l'estimée d'une image 3D+t ;

Concrètement, nous disposons de deux images 3D acquises à deux instants distincts : $I^0 = I(\mathbf{X}; 0)$ et $I^1 = I(\mathbf{X}; 1)$. Ces deux images bordent une séquence temporelle $I_{2D}(\mathbf{x}; t)$ décrivant les projections successives des volumes $I(\mathbf{X}; t)$ entre l'instant initial $t = 0$ et l'instant final $t = 1$ (voir figure 1). Notre étude méthodologique repose sur l'existence d'une fonctionnelle p qui réduit une scène réelle tri-dimensionnelle en une image bi-dimensionnelle. Ainsi, chaque image 2D de la séquence vidéo $I_{2D}(\mathbf{x}; t)$ est une projection de la structure 3D réelle correspondante. Cette fonctionnelle dépend uniquement du processus d'acquisition des données. Nous modélisons p par une transformation linéaire et stationnaire :

$$\begin{array}{rccc} p: & E & \longmapsto & F \\ & p(I)(x,y) & \longmapsto & \displaystyle\int_{\mathbb{R}} I(x,y,z) \underbrace{a(z)dz}_{d\mu(z)}, \end{array} \tag{1}$$

E correspond au domaine des images 2D et F à celui des images 3D et, la densité de mesure $a(z)$ dépend du dispositif d'acquisition et en particulier du plan de focalisation et $d\mu(z)$ est une mesure de Lebesgue. D'après cette formulation, cette projection intègre la contribution de tous les plans d'une structure 3D pour former une image 2D, la contribution de chaque plan est pondérée par la densité de mesure $a(z)$. La position de ce plan dépend des caractéristiques du dispositif d'acquisition et principalement de la distance focale. La transformation induit par conséquent une perte d'information importante dans la direction z. L'objectif est de réaliser une reconstruction 3D+t (temps/volume) complète de la scène observée à partir de deux échantillons I^0 et I^1 et d'une séquence vidéo $I_{2D}(\mathbf{x}, t)$. Les données 3D sont statiques, alors que la séquence $I_{2D}(\mathbf{x}, t)$, partiellement spatiale, présente une résolution temporelle intéressante. Nous souhaitons restituer une structure 3D à partir de chaque projection $I_{2D}(\mathbf{x}; t)$ existante. Évidemment, ce problème est mal posé puisqu'il existe une infinité d'images 3D correspondant à la projection 2D. Toutefois, nous pouvons contraindre l'espace des solutions en utilisant les valeurs des bords données par les images I^0 et I^1 et en imposant une contrainte de régularité. $\hat{I}(\mathbf{X}, t)$ désigne l'estimée tri-dimensionnelle déduite de $I_{2D}(\mathbf{x})$ à l'instant t. Une condition sur $\hat{I}(\mathbf{X}, t)$ découle directement du principe de reconstruction adopté. En effet, il s'agit de maximiser la ressemblance entre la projection de l'estimée 3D à l'instant t, *i.e.* $p(\hat{I})(\mathbf{X}, t)$, et la donnée $I_{2D}(\mathbf{x}, t)$. Cela nous fournit une première équation d'attache aux données qui contraint $\hat{I}(\mathbf{X}, t)$:

$$p(\hat{I})(\mathbf{x}, t) - I_{2D}(\mathbf{x}, t) = 0, \ \forall t \in]0\ , 1[\tag{2}$$

Puisque les instants t et $t-1$ sont supposés très proches et que les objets d'intérêt évoluent lentement pendant cette durée, nous pouvons mettre en correspondance $\hat{I}(\mathbf{X}, t)$ avec la structure 3D $I(\mathbf{X}, t-1)$ connue à l'instant $t-1$. Cette mise en correspondance est quantifiée en terme d'un champ de déplacement $\mathbf{W}$ entre les deux images. Il exprime l'hypothèse qu'un point en mouvement garde la même valeur de niveau de gris au cours du temps ce qui implique la contrainte de mouvement suivante :

$$\hat{I}(\mathbf{X}, t) = I(\mathbf{X} + \mathbf{W}, t-1) \tag{3}$$

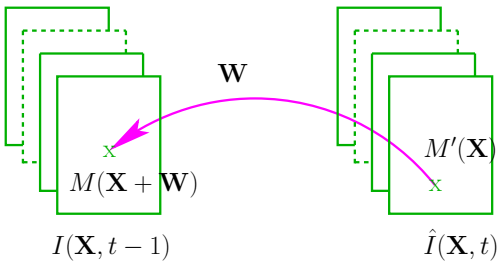


FIG. 2 – Mise en correspondance entre 2 structures 3D successives.

Nous estimons un champ rétrograde (voir figure 2) car la reconstruction des images par compensation de mouvement rétrograde simplifie le processus d'interpolation des voxels manquant dans les images estimées. Néanmoins, nous visualisons, dans la suite de l'article le champ opposé, *i.e.* $-\mathbf{W}$, traduisant le mouvement réel des pixels de $t-1$ à t. Puisque la distance temporelle entre deux acquisitions vidéo est infiniment petite, nous pouvons étendre le second membre de l'égalité précédente à un développement limité d'ordre 1. L'équation précédente s'écrit alors :

$$\hat{I}(\mathbf{X}, t) = \nabla I(\mathbf{X}, t-1).\mathbf{W} + I(\mathbf{X}, t-1), \tag{4}$$

où $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z})^T$ est l'opérateur gradient. Puisque p est linéaire, l'injection de l'équation (4) dans la contrainte d'attache aux données (2) produit une contrainte de données et de mouvement relative à $\mathbf{W}$:

$$p(\nabla I(\mathbf{X}, t-1).\mathbf{W}) + p(I(\mathbf{X}, t-1))(\mathbf{x}, t) - I_{2D}(\mathbf{x}, t) = 0 \tag{5}$$

Notre but est de résoudre l'équation (5) afin d'estimer un champ de déplacements $\mathbf{W}$ à chaque instant t. Nous présentons deux types d'approches d'estimation du

champ de vitesse de proche en proche. La première fournit un champ de vitesse C^0 par morceaux et la seconde estime un champ dense de vecteurs et régularisé.

En notant $I'_{2D}(\mathbf{x}, t)$ la différence d'images $p(I(\mathbf{X}, t-1)) - I_{2D}(\mathbf{x}, t)$, l'équation (5) s'écrit plus simplement :

$$p(\nabla I(\mathbf{X}, t-1).\mathbf{W}) + I'_{2D}(\mathbf{x}, t) = 0 \tag{6}$$

Cette équation est de type flot optique dans laquelle le gradient temporel de l'image est remplacé par une différence d'image projetée.

Notre objectif est de résoudre cette équation pour chaque structure 3D, $I(\mathbf{X}, t-1)$, et chaque différence d'images, $I'_{2D}(\mathbf{x}, t)$, afin de reconstruire la séquence de proche en proche.

4 Résolution directe par approximation constante par morceaux

Le principe consiste à résoudre l'équation (6) sans recourir à l'ajout d'une contrainte de régularisation supplémentaire sur le champ de vitesse recherché. Le modèle ajuste, dans chaque voisinage spatial Ω_l du point considéré (voir figure 3), les contraintes locales du premier ordre à une valeur fixe de $\mathbf{W}_l$. Cela revient à résoudre l'équation (6) en tout point de l'image volumique en supposant que le champ soit constant en tout point d'un voisinage donné centré autour de ce point [9]. Ainsi, pour estimer le champ de vitesse en une position donnée de la structure 3D, il suffit de définir un voisinage Ω_l cubique de côté n et de résoudre localement l'équation (6) sous l'hypothèse que le champ de vecteur vitesse est égal à une valeur fixe $\mathbf{W}_l$ sur Ω_l. Cela se traduit par l'équation suivante valable sur le sous-domaine Ω_l :

$$E(\mathbf{W}_l) = \sum_{i=-\frac{n}{2}}^{\frac{n}{2}} \sum_{j=-\frac{n}{2}}^{\frac{n}{2}} \|p_k(\nabla I.\mathbf{W}_l)(i,j) + I'_{2D}(i,j)\|^2 \tag{7}$$

où $\mathbf{W}_l$ est un vecteur constant dans Ω_l, J_k la restriction de $I(\mathbf{X}, t-1)$ au voisinage considéré de côté n et p_k la restriction de la projection p aux plans de ce voisinage, définie par :

$$p_k(I)(i,j) \quad = \quad \sum_{m=k-\frac{n}{2}}^{m=k+\frac{n}{2}} a_m J_k(i,j,m), \tag{8}$$

Les coefficients a_m proviennent de la formulation discrète de la projection p donnée, dans le cas continu, par l'équation (1) à la section 3. Ainsi on peut écrire matriciellement E sous la forme :

$$E(W) = \|PAW - B\|^2, \tag{9}$$

Quatre éléments agissent dans l'énergie E : P, A, W et B se rapportant respectivement à la projection, la fenêtre cubique des gradients spatiaux, le vecteur vitesse et la composante temporelle. $P(n \times n, n \times n \times n)$ est une matrice bande (voir équation (10) à la page suivante) dont les éléments dépendent des coefficients a_k de la restriction de la projection p_k au voisinage considéré.

$$P = \begin{pmatrix} a_{k-\frac{n}{2}} \dots a_{k+\frac{n}{2}} & 0 & \dots & & 0 \\ 0 \dots\dots\dots 0 & a_{k-\frac{n}{2}} \dots a_{k+\frac{n}{2}} & 0 & \dots & 0 \\ \vdots & 0 \dots\dots\dots 0 & & \ddots & 0 \\ \vdots & \dots & \dots & 0 & a_{k-\frac{n}{2}} \dots a_{k+\frac{n}{2}} \end{pmatrix} \tag{10}$$

Pour un voisinage centré au pixel P de coordonnée (i, j, k), chaque colonne de A (voir équation (12) représente une suite ordonnée de pixels obtenue en parcourant d'abord les restrictions des gradients spatiaux I_x, I_y et I_z à ce voisinage en profondeur puis en hauteur et finalement en largeur. Le voisinage Ω_l est représenté schématiquement à la figure 3. Rappelons que :

$$\nabla I = (I_x, I_y, I_z)^T \tag{11}$$

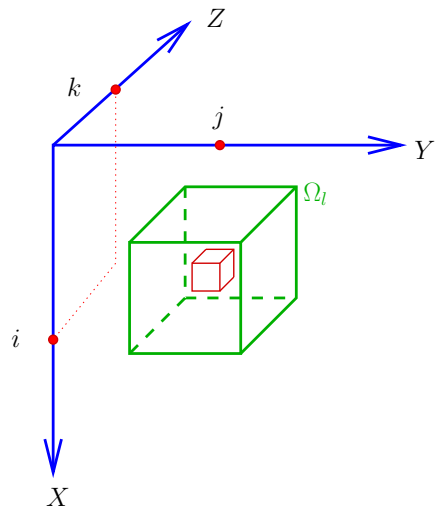


FIG. 3 – Représentation schématique du voisinage Ω_l.

$$A(n \times n \times n, 3) = \begin{pmatrix} \nabla I(i-\frac{n}{2}, j-\frac{n}{2}, k-\frac{n}{2}) \\ \nabla I(i-\frac{n}{2}, j-\frac{n}{2}, k-\frac{n}{2}+1) \\ \vdots \\ \nabla I(i-\frac{n}{2}, j-\frac{n}{2}, k+\frac{n}{2}) \\ \nabla I(i-\frac{n}{2}+1, j-\frac{n}{2}, k-\frac{n}{2}) \\ \vdots \\ \nabla I(i-\frac{n}{2}+1, j-\frac{n}{2}, k+\frac{n}{2}) \\ \vdots \\ \vdots \\ \nabla I(i+\frac{n}{2}, j-\frac{n}{2}, k+\frac{n}{2}) \\ \nabla I(i+\frac{n}{2}, j-\frac{n}{2}+1, k-\frac{n}{2}) \\ \vdots \\ \nabla I(i+\frac{n}{2}, j-\frac{n}{2}+1, k+\frac{n}{2}) \\ \vdots \\ \vdots \\ \vdots \\ \nabla I(i+\frac{n}{2}, j+\frac{n}{2}, k+\frac{n}{2}) \end{pmatrix} \quad (12)$$

$\mathbf{W}_l(3,1)$ est un vecteur constant à trois composantes, soit :

$$W_l = \begin{pmatrix} u_l \\ v_l \\ w_l \end{pmatrix}$$

Et $B(n \times n)$ est un vecteur représentant une suite ordonnée des pixels composant le voisinage carré de côté n considéré dans la différence d'images I'_{2D} :

$$B = \begin{pmatrix} I'_{2D}(i - \frac{n}{2}, j - \frac{n}{2}) \\ I'_{2D}(i - \frac{n}{2}, j - \frac{n}{2} + 1) \\ \vdots \\ I'_{2D}(i - \frac{n}{2}, j + \frac{n}{2}) \\ I'_{2D}(i - \frac{n}{2} + 1, j - \frac{n}{2}) \\ \vdots \\ I'_{2D}(i - \frac{n}{2} + 1, j + \frac{n}{2}) \\ \vdots \\ \vdots \\ I'_{2D}(i + \frac{n}{2}, j + \frac{n}{2}) \end{pmatrix}$$

Le problème de la minimisation de (9) correspond donc simplement à une régression linéaire. Dans ce cas, la solution est directe et donnée par la méthode des moindres carrés [2] :

$$W_l = [A'^T A']^{-1} A'^T B, \tag{13}$$

Avec $A' = PA$ et A'^T la transposée de A'. La matrice produit $A'(n \times n, 3)$ a la forme analytique suivante :

$$A' = \begin{pmatrix} \sum_{m=k-\frac{n}{2}}^{m=k+\frac{n}{2}} a_m \nabla I(i-\frac{n}{2}, j-\frac{n}{2}, m) \\ \sum_{m=k-\frac{n}{2}}^{m=k+\frac{n}{2}} a_m \nabla I(i-\frac{n}{2}, j-\frac{n}{2}+1, m) \\ \vdots \\ \sum_{m=k-\frac{n}{2}}^{l=k+\frac{n}{2}} a_m \nabla I(i-\frac{n}{2}, j+\frac{n}{2}, m) \\ \vdots \\ \vdots \\ \sum_{m=k-\frac{n}{2}}^{m=k+\frac{n}{2}} a_m \nabla I(i+\frac{n}{2}, j+\frac{n}{2}, m) \end{pmatrix}$$

L'équation (9) peut être résolue si la matrice $A'^T A'$ n'est pas singulière, ce qui n'est *a priori* jamais le cas sauf pour les zones où la distribution de l'intensité est constante. $A'^T A'$ est la matrice de taille 3×3 symétrique définie par l'expression donnée par l'équation (14) à la page suivante.

$$A'^T A' = \begin{pmatrix} \sum_{n\times n} p_k(I_x)^2 & \sum_{n\times n} p_k(I_x)p_k(I_y) & \sum_{n\times n} p_k(I_x)p_k(I_z) \\ \sum_{n\times n} p_k(I_x)p_k(I_y) & \sum_{n\times n} p_k(I_y)^2 & \sum_{n\times n} p_k(I_y)p_k(I_z) \\ \sum_{n\times n} p_k(I_x)p_k(I_z) & \sum_{n\times n} p_k(I_y)p_k(I_z) & \sum_{n\times n} p_k(I_z)^2 \end{pmatrix} \quad (14)$$

Cette méthode permet en principe de fournir un champ dense de vecteur vitesse puisque en chaque point l de l'image, on peut calculer un vecteur vitesse par inversion de l'équation du flot optique sur le voisinage Ω_l. En pratique, on ne procède pas ainsi car les coûts algorithmiques seraient trop importants. Il est préférable de calculer le champ de vitesse sur une grille sous-échantillonnée. On peut alors

faire se chevaucher les voisinages Ω_l pour contrôler dans une certaine mesure la régularité spatiale du champ des vitesses.

5 Approche variationnelle

Une seconde méthode proposée est d'estimer un champ dense par formulation variationnelle [6]. Afin de résoudre l'équation (6), nous construisons une énergie dont le minimum fournit une estimation de la solution. Le terme de coût lié à cette contrainte d'attache aux données est :

$$E_1(\mathbf{W}) = \iint_{\mathbb{R}^2} \big(p(\nabla I(\mathbf{X}, t-1).\mathbf{W}) + I'_{2\mathrm{D}}(\mathbf{x}, t)\big)^2 dxdy$$

En introduisant les gradients spatiaux de $I(\mathbf{X}, t-1)$ dans les trois directions (I_x, I_y et I_z) (11), cette équation s'écrit :

$$E_1(\mathbf{W}) = \iint_{\mathbb{R}^2} \big(p(I_x u + I_y v + I_z w) + I'_{2\mathrm{D}}(\mathbf{x}, t)\big)^2 dxdy \tag{15}$$

La solution du problème de la minimisation de (15) est alors donnée par un zéro des équations d'Euler-Lagrange associées, c'est-à-dire $\dfrac{\partial E_1}{\partial \mathbf{W}} = 0$. La quantité $\dfrac{\partial E_1}{\partial \mathbf{W}}$ est la différentielle de la fonctionnelle E_1. Cette fonctionnelle est convexe, donc les équations d'Euler-Lagrange associées ont une solution unique qui correspond à un minimum global de (15). Malheureusement l'équation (6) est sous dimensionnée car le vecteur $\mathbf{W}$ possède 3 composantes. Nous ajoutons, dans la fonctionnelle, un terme régularisant qui contraint les variations spatiales de $\mathbf{W}$:

$$E_2(\mathbf{W}) = \iiint_{\mathbb{R}^3} \Big(\|\nabla u\|^2 + \|\nabla v\|^2 + \|\nabla w\|^2 \Big) dxdydz$$

Finalement, nous minimisons l'énergie totale en $\mathbf{W}$:

$$E(\mathbf{W}) = E_1(\mathbf{W}) + \alpha E_2(\mathbf{W}), \tag{16}$$

où α est un coefficient pondérant les effets de la régularisation. Nous avons maintenant un problème bien posé mais la nouvelle fonctionnelle n'est plus convexe. Cela implique que la solution obtenue par calcul d'un zéro de la différentielle de E correspond à un minimum local.

Nous devons maintenant calculer explicitement la différentielle de E. Pour cela nous utilisons la dérivée directionnelle définie par la formule de Gâteau :

Définition 1 (Dérivée de Gâteau) *La dérivée d'une fonctionnelle E en un point u dans la direction g est donnée par :*

$$\lim_{\lambda \to 0} \frac{F(u + \lambda g) - F(u)}{\lambda}$$

Si cette quantité existe, c'est une forme linéaire continue en g et nous la notons $\frac{\partial F}{\partial u}(g)$.

Propriété 1 *Dans ce cas, le théorème de représentation de Riesz établit que :*

$$\frac{\partial F}{\partial u}(g) = \langle \frac{\partial F}{\partial u}, g \rangle_{L^1}, \tag{17}$$

où $\langle , \rangle_{L^1}$ est le produit scalaire associé à l'espace L^1 des fonctions intégrables.

Le terme $\dfrac{\partial F}{\partial u}$ est la différentielle recherchée. Nous devons donc écrire les dérivées directionnelles sous la forme d'un produit scalaire. Comme $\mathbf{W}$ est une fonction à valeurs dans $\mathbb{R}^3$, nous effectuons d'abord le calcul des dérivées de E_1 par rapport à u, v et w fixes, suivant une direction g quelconque. En d'autres termes, il s'agit de calculer :

$$\lim_{\lambda \to 0} \frac{E_1(u + \lambda g) - E_1(u)}{\lambda} \tag{18}$$

Nous développons alors l'expression $E_1(u + \lambda g, v, w) - E_1(u, v, w)$ et obtenons après simplification :

$$\begin{array}{rcl} E_1(u + \lambda g, v, w) - E_1(u, v, w) & = & \\ \lambda^2 \iint p(I_x g)^2 dxdy & + & \\ 2\lambda \iint p(I_x g)[p(I_x u + I_y v + I_z w) + I'_{2\mathrm{D}}]dxdy & & \end{array} \tag{19}$$

En divisant par λ et en passant à la limite, nous obtenons :

$$\begin{array}{rcl} \lim\limits_{\lambda \to 0} \dfrac{E_1(u + \lambda g) - E_1(u)}{\lambda} & = & \\ 2 \iint p(I_x g)[p(I_x u + I_y v + I_z w) + I'_{2\mathrm{D}}]dxdy & & \end{array} \tag{20}$$

À cause de la présence de la fonction de projection p, le second membre de l'égalité (20) n'est pas sous la forme du produit scalaire défini par l'expression (17).

Nous remplaçons, dans l'équation (20), p par son expression générale dans le cas continu 1. Injecté dans le premier membre de l'égalité (20), nous obtenons :

$$\begin{array}{l} \iint p(I_x g)\big(p(I_x u + I_y v + I_z w) + I'_{2\mathrm{D}}\big)dxdy \quad = \\ \iint \left[\int_z a(z) I_x g dz\right] \big(p(I_x u + I_y v + I_z w) + I'_{2\mathrm{D}}\big)dxdy \end{array} \tag{21}$$

La fonction $\psi(x,y) = p(I_x u + I_y v + I_z w) + I'_{2\mathrm{D}}$ dépend uniquement des variables x et y. En d'autres termes, à cause de la présence de la fonction de projection p, cette fonction est indépendante de la variable z. D'après le théorème de Fubini, nous avons :

$$\begin{array}{l} \iint p(I_x g)\psi(x,y)dxdy \quad = \\ \iiint a(z) I_x g(x,y,z).\psi(x,y)dzdxdy \end{array} \tag{22}$$

En posant le changement de mesure : $a_z dz = d\mu(z)$, l'égalité (22) devient :

$$\begin{array}{l} \iint p(I_x g)\psi(x,y)dxdy \quad = \\ \iiint I_x g(x,y,z)\psi(x,y)dxdyd\mu(z) \end{array} \tag{23}$$

Donc :

$$\begin{array}{l} \lim\limits_{\lambda\to 0} \dfrac{E_1(u+\lambda g) - E_1(u)}{\lambda} = \\ 2\iiint I_x g\big[p(I_x u + I_y v + I_z w) + I'_{2\mathrm{D}}\big]dxdyd\mu(z) \end{array} \tag{24}$$

Nous remarquons alors que cette dernière expression représente bien le produit scalaire entre deux fonctions données définies en (17). Ainsi :

$$\lim_{\lambda\to 0} \frac{E_1(u+\lambda g) - E_1(u)}{\lambda} = \langle g, 2I_x\big[p(I_x u + I_y v + I_z w) + I'_{2\mathrm{D}}\big]\rangle \tag{25}$$

D'après la propriété 1, la dérivée de Gâteau par rapport à u est donc :

$$\frac{\partial E_1}{\partial u} = 2I_x\big[p(I_x u + I_y v + I_z w) + I'_{2\mathrm{D}}\big] \tag{26}$$

D'une manière analogue, les différentielles de E_1 en v et w sont :

$$\begin{aligned}\frac{\partial E_1}{\partial v} &= 2I_y\big[p(I_x u + I_y v + I_z w) + I'_{2D}\big] \\ \frac{\partial E_1}{\partial w} &= 2I_z\big[p(I_x u + I_y v + I_z w) + I'_{2D}\big]\end{aligned}$$

Nous ne donnons pas les détails du calcul des gradients du terme de régularisation E_2 qui est classique, le lecteur pourra consulter [4, 12]. Ce calcul donne le système d'équations suivant :

$$\begin{aligned}\frac{\partial E_2}{\partial u} &= -2\alpha\Delta u \\ \frac{\partial E_2}{\partial v} &= -2\alpha\Delta v \\ \frac{\partial E_2}{\partial w} &= -2\alpha\Delta w\end{aligned}$$

Nous écrivons ensuite que les trois composantes de $\dfrac{\partial E}{\partial \mathbf{W}}$ sont nulles pour aboutir au système d'équations suivant :

$$\begin{aligned} I_x\big[p(I_x u + I_y v + I_z w) + I'_{2D}\big] - \alpha\Delta u = 0 \\ I_y\big[p(I_x u + I_y v + I_z w) + I'_{2D}\big] - \alpha\Delta v = 0 \\ I_z\big[p(I_x u + I_y v + I_z w) + I'_{2D}\big] - \alpha\Delta w = 0\end{aligned} \tag{27}$$

Afin de converger vers un minimum local, nous utilisons les équations d'évolution associées à celles du système (27). Nous sommes toutefois assurés d'atteindre le minimum global que si la condition initiale est suffisamment proche de la solution. Les équations d'évolution sont alors équivalentes à une méthode de descente dans la direction du gradient :

$$\begin{aligned}\frac{du}{dt} + I_x\big[p(I_x u + I_y v + I_z w) + I'_{2D}\big] - \alpha\Delta u = 0 \\ \frac{dv}{dt} + I_y\big[p(I_x u + I_y v + I_z w) + I'_{2D}\big] - \alpha\Delta v = 0 \\ \frac{dw}{dt} + I_z\big[p(I_x u + I_y v + I_z w) + I'_{2D}\big] - \alpha\Delta w = 0\end{aligned}$$

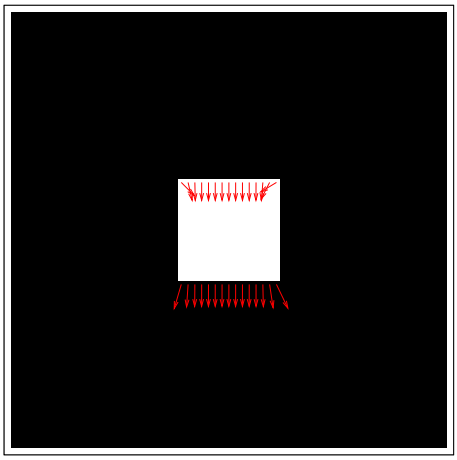
6 Expérimentation

Afin de valider l'étude théorique concernant les approches ne tenant pas compte d'un modèle de données bien déterminé, nous construisons un lot de données volumiques. Nous considérons une séquence 3D+t simple contenant un cube de couleur homogène qui se déplace vers le bas et en profondeur, vers l'intérieur. Nous simulons la projection perspective produisant la séquence vidéo 2D+t intermédiaire. Considérons maintenant la première image 3D I^0 obtenue et la première projection $I_{2D}(\mathbf{x}, t_1)$. Ces deux structures nous permettent de calculer le champ de déplacements entre les deux structures et de restituer la donnée 3D à l'instant t_1. Nous estimons ce champ de déplacement à l'aide des deux méthodes présentées dans cet article. Afin de visualiser le résultat de calcul des champs de déplacements, nous représentons une section de l'image volumique avec la restriction des champs calculés aux composantes u et v (voir figure 4). Nous rappelons que les composantes u et v contiennent des informations relatives aux déplacements dans la direction respectivement de X et de Y. Enfin, les gradients spatio-temporels qui constituent nos données d'entrée sont calculés par convolution sur un noyau gaussien 3D dérivé respectivement dans les trois directions. La variance du noyau est fixée à 1 ce qui correspond à un lissage modéré.

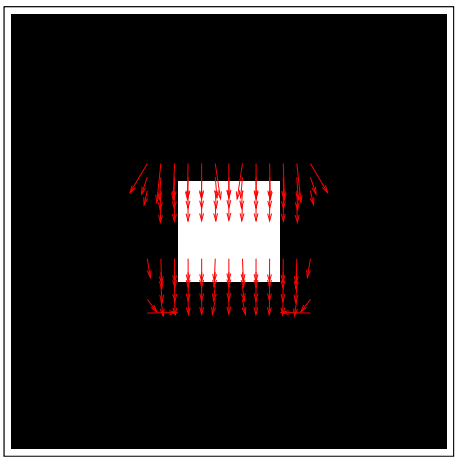
D'après la figure 4, le champ de déplacements généré par la méthode différentielle globale est lisse. De plus, les déplacements sont localisés sur les frontières du cube en mouvement. Néanmoins, les déplacements estimés par la méthode locale s'étalent sur une région plus large. Cela est dû au fait que l'estimation locale du mouvement sur un voisinage de taille donnée produit un champ continu par morceaux. Plus la taille du voisinage considéré est grande, meilleure est la qualité de reconstruction, mais plus lente est la résolution. Nous abordons, par ailleurs, un cas d'occultation d'objets en profondeur. Nous considérons une image 3D contenant deux cubes qui se recouvrent lors de la projection 2D. D'après la figure 5, nos deux méthodes estiment des champs corrects de point de vue amplitude et directions.

7 Conclusion

Nous avons présenté dans cette étude deux approches possibles pour résoudre la problématique de la reconstruction 3D+t n'impliquant pas de connaissances *a priori* sur le modèle de données. Le principe est de résoudre une équation de transport de la luminosité en chaque voxel, en établissant des hypothèses sur la

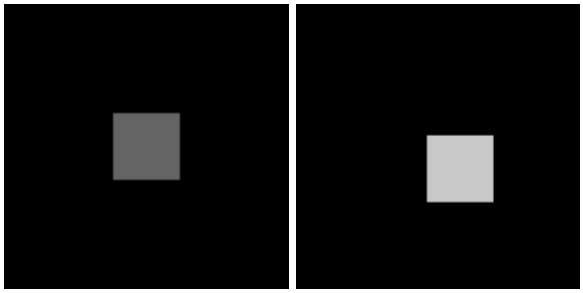


(a) Champ de vitesses estimé à l'aide de la méthode différentielle globale, $\alpha = 100$, le nombre d'itérations est de 50.

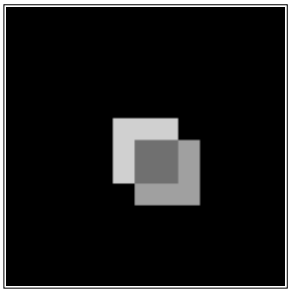


(b) Champ de vitesses estimé à l'aide de la méthode locale, testé sur un voisinage cubique de côté 5 pixels.

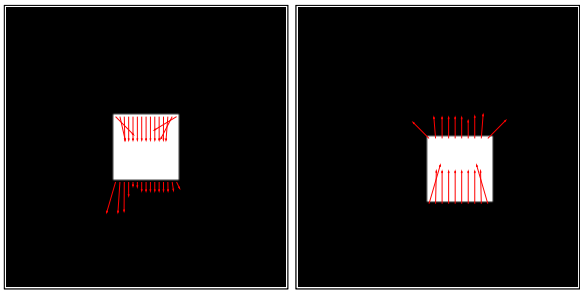
FIG. 4 – Représentation des composantes u et v des champs de vitesse estimés, de gauche à droite, par la méthode variationnelle (a) et la méthode directe (b) sur une section d'une image volumique comprenant un cube qui se déplace de haut vers le bas et vers l'intérieur.



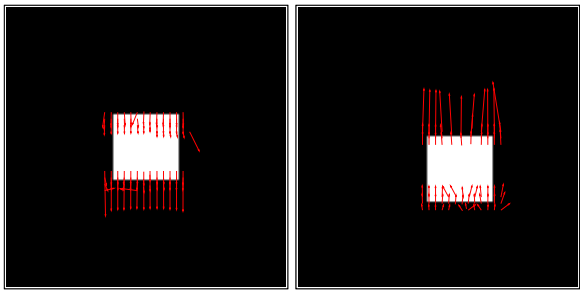
(a) Coupe de l'image 3D I^0 à deux profondeurs montrant les deux carrés en déplacement.



(b) Image 2D $I_{2D}(\mathbf{x}, t_1)$, projection à l'instant t_1.



(c) Champ de vitesse estimé à l'aide de la méthode différentielle, $\alpha = 100$, le nombre d'itérations est de 50.



(d) Champ de vitesse estimé à l'aide de la méthode directe, testé sur un voisinage cubique de côté 5 pixels.

FIG. 5 – Représentation des composantes u et v des champs de vitesse estimés, de haut en bas, par la méthode variationnelle et la méthode directe sur deux sections d'une image volumique comprenant deux cubes qui se déplacent respectivement vers le bas et vers le haut.

nature du mouvement à calculer. La méthode différentielle consiste à contraindre spatialement le champ de déplacement estimé de proche en proche. Cette méthode génère un champ dense ayant un aspect visuel lisse satisfaisant. La méthode directe consiste à résoudre l'équation de conservation de la luminosité en supposant que le champ de vitesse soit constant sur un voisinage de taille donnée. Cette méthode fournit un champ continu par morceaux. Plus la taille du voisinage considéré pour la résolution de cette équation est élevée, meilleure est la qualité de reconstruction, mais aussi, plus le temps d'exécution de l'algorithme associé est important. Ce dernier, comparé à la résolution itérative mise en œuvre dans la méthode différentielle, présente l'avantage d'être plus simple. En évitant le lissage généré par la régularisation, pouvant être une contrainte trop forte sur les données, il peut s'avérer plus adapté au contexte des données biologiques. Néanmoins, l'exécution de cet algorithme peut induire des erreurs d'estimation des déplacements car elle repose sur l'inversion de matrices dépendant des gradients de l'image. En particulier la méthode ne fonctionne pas sur des zones homogènes. Cette difficulté peut être levée en excluant du calcul les zones de pixels de faible norme du gradient spatial. Nous avons montré que la reconstruction 3D est possible par calcul du flot optique. Évidemment, ces approches souffrent des limitations connues de la formulation relative à la contrainte du flot optique. En l'occurrence, l'estimation des larges déplacements n'est pas adapté. Cette formulation a aussi une faible robustesse vis à vis des gradients spatio-temporels servant de données d'entrée. De plus, la régularisation L^2 mise en œuvre dans l'approche variationnelle peut être très contraignante. Néanmoins, il est parfaitement possible d'adapter toutes les variantes qui existent autour de la résolution du flot optique qui permettent de résoudre les problèmes que nous venons de citer.

Références

[1] P.J. Besl and N. McKay. A method for registration of 3-d shapes. *IEEE PAMI*, 14(2) :239–256, Mar. 1992.

[2] R. Burden and L. Fairies. *Numerical Analysis*. Prindle, Weber & Schmidt, 1985.

[3] G. Demoment. Image reconstruction and resotration : overview of common estimation structures and problems. *IEEE Trans. on ASSP*, 37(12) :2024–2036, Dec. 1989.

[4] R. Glowinski. *Numerical Methods for Nonlinear Variational Problems*. Series in computational physics, New York, springer edition, 1984.

[5] A. Gueziec, P. Kazanzides, B. Williamson, and RH. Taylor. Anatomy-based registration of CT-scan and intraoperative X-ray images for guiding a surgical robot. *IEEE Trans Med Imaging*, 17 :715–728, 1998.

[6] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17 :185–203, 1981.

[7] L. Huei-Yung. *Computer vision techniques for complete 3D model reconstruction*. PhD thesis, State university of New York at Stony Brook, 2002.

[8] L. Lemieux, R. Jagoe, D.R. Fish, N.D. Kitchen, and D.G. Thomas. A patient-tocomputed-tomography image registration method based on digitally reconstructed radiographs. *Med Phys*, 21 :1749–1760, 1994.

[9] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 674–697, Canada, 1981.

[10] J.B.A. Maintz and M.A. Viergever. A survey of medical image registration. *Medical Image Analysis*, 2(1), 1998.

[11] F. Natterer. *The mathematics of computerized tomography*. Wiley, New York, 1986.

[12] W. Rekik. *Fusion de données temporelles, ou 2D+t, et spatiales, ou 3D, pour la reconstruction de scènes 3D+t et traitement d'images sphériques. Applications à la biologie cellulaire*. PhD thesis, Université Pierre Et Marie Curie, Paris 6, Département Calcul Scientifique, Laboratoire d'Informatique de Paris 6 (LIP6), février 2007.

[13] W. Rekik, D. Béréziat, and S. Dubuisson. 3D+t reconstruction in the context of locally spheric shaped data observation. In *12th International Conference on Computer Analysis of Images and Patterns*, Vienna, Austria, 2007. Accepted for publication.

[14] P. Viola and W. Wells. alignment by maximization of mutual information. *International Journal of Computer Vision*, 24 :137–154, 1997.

3D+t Reconstruction in the Context of Locally Spheric Shaped Data Observation.

Wafa Rekik[1], Dominique Béréziat[2], and Séverine Dubuisson[1]

[1] Université Pierre et Marie Curie (UPMC),
Laboratoire d'Informatique de Paris 6 (LIP6)
104 Avenue du Président Kennedy, 75016 Paris
[2] Clime project/INRIA Rocquencourt
B.P. 105 78153 Le Chesnay Cedex France
Wafa.Rekik@lip6.fr

Abstract. The main focus of this paper is 3D+t shape recovery from 3D spatial data and 2D+t temporal sequences. This reconstruction is particularly challenging due to the great deal of in-depth information loss observed on the 2D+t temporal sequence. Our approach embed a geometrical local constraint to handle the critical lack of information. This prior constraint is defined by a spherical topology because several applications may be concerned. It allows us to model relevantly the 3D-to-2D transformation that reduces each 3D image into a 2D frame. We then can build a 3D inaccurate inverse reconstruction of each 2D frame belonging to the video, *i.e.* 2D+t sequence. These inaccurate 3D images are enhanced by gradual motion compensation using a regularity criterion. Results on synthetic data are displayed.

Key words: 3D reconstruction, motion compensation, variational formulation, optical flow constraint

1 Introduction

In some computer vision applications, we deal with volumetric, *i.e* 3D, acquisitions framing temporal, *i.e* 2D+t, ones (figure 1). First type of acquisitions provides a description of the 3D scene geometry, thus, purely spatial information. Second ones observe 2D object motion providing temporal and partial spatial kind of information. A possible approach to mine exhaustively these complementary datasets is to carry out a complete spatio-temporal or a 3D+t scene reconstruction. 3D+t sequences are restored by recovering the 3D original volume from each 2D frame belonging to the 2D+t sequence. As the matter of fact, 3D reconstruction is an inverse problem. In this particular case, it is also an ill posed problem since a single 2D frame is hardly sufficient to recover the 3D original volume. Indeed, this frame exhibits a great deal of spatial distortion and in-depth loss of information caused by the projective transformation that reduces the 3D real structure into a 2D image. To handle this critical lack of information, we introduce a prior data model. It consists in a geometrical constraint

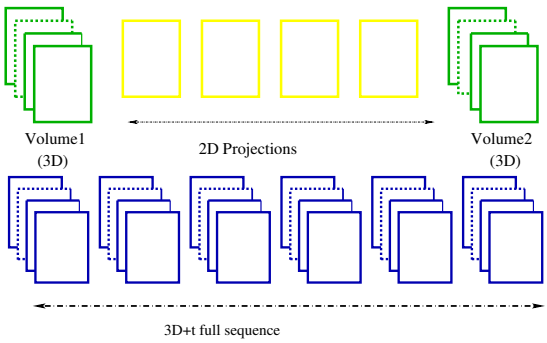


Fig. 1. Recovering a 3D+t sequence from a couple of 3D images framing a series of 2D projections.

defined a spherical topology. Data observing locally spheric shaped objects can be issued from remote sensing acquisitions and especially meteorology, from astronomical ones focusing on the sun and from bio-cellular microscopic imagery. A relevant biologic application is described in [1] where biologists study dynamic of structures of interest evolving on a spherical cell simulation surface. In this context, for instance, spatial datasets consist in multifocus acquisitions. Hence, they are not actual 3D images since they are formed by a series of equal sized slices, where signal in each slice is bothered by luminosity coming from adjacent ones. A spatial deconvolution stage is therefore needed when biologists calibrate correctly the video microscopy system to cell wall simulation experiments. Due to this limitation and to respect generality, we use synthetic data to validate implemented algorithms.
Our approach uses this geometrical prior constraint in order to model relevantly the 3D-to-2D transformation. This modeling allows us to build a 3D inaccurate inverse reconstruction of each 2D frame belonging to the video sequence. These inaccurate 3D images are enhanced by gradual motion compensation using a regularity criterion. We proceed with a short overview of 3D reconstruction methods using multiple/single projective views in section 2. Then, we move to the description of our 3D+t reconstruction approach in section 3. Experimental results on synthetic datasets are given in section 4.

2 State-of-the-art

In the literature, a wide scope of approaches addresses the 3D reconstruction problem. Volume based methods aim at restoring an accurate description of luminosity in each voxel of the 3D reconstructed image, while surface based ones provide a 3D model of the object of interest. This model requires shape and texture recovery of the outer visible side of the observed object. Namely, tomographic reconstruction [2] focuses on restoring the original volume from multiple projective views. Stereo-vision [3] approaches make use in general of the tri-

angulation principle in order to recover the surface shape of selected targets. Moreover, 2D-to-3D registration methods reconstruct 3D features from only one projective view. This registration consists in identifying a geometrical transformation that aligns 2D frames with 3D datasets often acquired with a different modality. This alignment determines their relative position and orientation and then locates them in the coordinate system of the 3D images. It is of growing interest in the medical imaging field mainly to register 3D pre-operative images (like MRI, CT images) into 2D series of intra-operative frames acquired during a surgical intervention. Registration may operate on feature-based methods or intensity based ones. Reader is referred to [4, 5]. Tomography and stereovison, namely, require multiple projective views to achieve the reconstruction task. Therefore, aligning 2D projection images to 3D datasets approaches appear to be adapted to our context of study. However they fit into a registration framework rather than a reconstruction one, unless other projective views are available, which is quite unusual. They can be used alternatively for a preprocessing stage for some aforementioned applications. We built then an adapted reconstruction model involving a prior geometrical constraint.

3 3D+t reconstruction approach

We carry out the 3D+t reconstruction using a couple of 3D data acquired at two different instants as well as the 2D+t sequence filmed meanwhile. Since the intermediary video sequence presents an interesting temporal resolution, we propose to restore the underlying 3D structure of each frame. The latter results from a reduction of the real 3D structure into a 2D projective view at a given time t. We take into account the prior geometrical constraint related to the observed object in order to build an inaccurate 3D inverse reconstruction of each 2D frame. We then attempt to improve this inaccurate 3D representation by motion compensation. We compute a motion vector field matching the 3D inaccurate structure with the 3D one already estimated correctly at time $t-1$. The whole 3D+t sequence is reconstructed gradually using the couple of 3D data as initial border conditions.

3.1 Modeling assumption

We carry out the 3D+t reconstruction using a series of 2D+t video sequence acquisitions intersected with 3D volumetric ones. We opt for the following notations for the remainder of the paper: $\mathbf{X} = (x, y, z)$, $\mathbf{x} = (x, y)$, $\mathbf{W} = (u, v, w)$, $I(\mathbf{X}, t)$, $I_{2D}(\mathbf{x}, t)$ and $\hat{I}(\mathbf{X}, t)$ are respectively a 3D vector position, a 2D vector position, a 3D vector field, a 3D+t sequence, a 2D+t one and finally the estimate of a 3D image at time t. Concretely, we have two 3D structures at two different instants: $I^0 = I(\mathbf{X}; t_0)$ et $I^1 = I(\mathbf{X}; t_1)$. Between $t = t_0$ and $t = t_1$, we also have a video sequence $I_{2D}(\mathbf{x}; t)$ describing all the 2D reductions of the $I(\mathbf{X}; t)$ structures evolving in this duration, *i.e.* between $t = t_0$ and $t = t_1$. Each 2D frame I_{2D} is a projective sight, at a given instant t, of the 3D original volume. This projection

depends only on the data acquisition mechanism. We model it by a linear and stationary transformation, called projection p: $p(I)(x,y) = \int_{\mathbb{R}} I(x,y,z)h(z)dz$, where $h(z)$ is a measure kernel describing the interaction between observed object and the input light signal. p integrates contributions of all 3D slices to produce an unique 2D image. It introduces then, an important loss of information in the $z-$direction.

3.2 Prior geometrical constraint

Let us point out that, in our case of study, structures of interest evolve on a spheroid surface. The transformation p does not introduce distortions on respectively x and $y-$directions, consequently, p is reduced merely to a sum of two contributions coming from, respectively, the frontal and the dorsal hemisphere. 2D frames suffer then from a positioning ambiguity. Indeed, we are unable to assert that a projected object belongs either to the frontal or dorsal hemisphere. We suppose that the spherical support has a constant radius that we estimated in a pre-processing stage. We map the texture of each 2D frame on both sides of a sphere of same parameters that the original surface. This leads to an inaccurate 3D+t sequence $I_G(\mathbf{X},t)$ geometrically similar to the original 3D+t sequence $I(\mathbf{X},t)$, but holding errors generated by the positioning ambiguity (figure 2). The sequence I_G is formally obtained by $I_G \circ G(\mathbf{x},t) = I_{2D}(\mathbf{x},t)$ where G is defined by $G(x,y,z) = (x,y)^T$ if $(x-c_x)^2+(y-c_y)^2+(z-c_z)^2 = R^2$. Parameters (c_x,c_y,c_z) and R are respectively the center and the radius of the sphere surface.

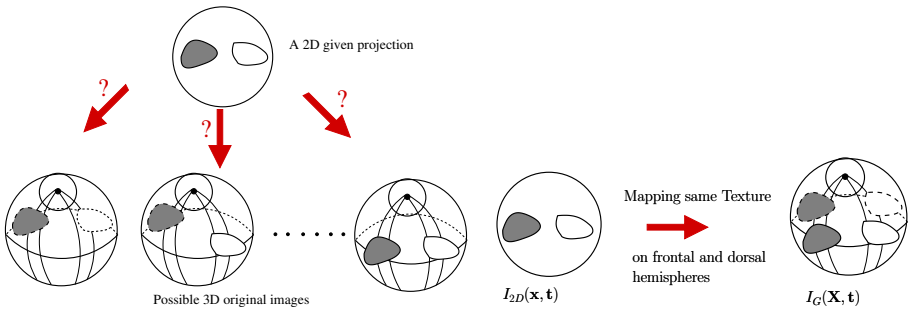


Fig. 2. From left to right: illustration of the positioning ambiguity problem and recovering inaccurate 3D images, $I_G(\mathbf{X},t)$, from 2D frames $I(\mathbf{x},t)$.

Building the inaccurate 3D+t sequence yields the following data model equation:

$$\hat{I}(\mathbf{X},t) = I_G(\mathbf{X},t), t \in]t_0;t_1[\quad (1)$$

3.3 Motion compensation

In order to restore correctly each luminosity contribution in the corresponding hemisphere, we match the inaccurate 3D representation, $I_G(\mathbf{X},t)$, with the 3D

image available at $t-1$, $I(\mathbf{X}, t-1)$. This matching is quantified as a displacement vector field $\mathbf{W}$ describing voxel motion between both structures $I_G(\mathbf{X}, t)$ and $I(\mathbf{X}, t-1)$. Since times t and $t-1$ are assumed to be very close and objects of interest evolve scarcely in this duration, we make the assumption that a moving voxel keeps the same gray value over time. A straightforward equation, similar to optical flow constraint, is then issued under the constant brightness assumption from time t to $t-1$:

$$\hat{I}(\mathbf{X}, t) = I(\mathbf{X} + \mathbf{W}, t-1) \tag{2}$$

We recall that $\hat{I}(\mathbf{X}, t)$ accounts for the estimate of the 3D image at time t. We estimate a retrogress vector field defining voxel displacement from $t-1$ to t for notation conveniences. Expanding the right-hand side of equation (2) in a first order Taylor series leads to: $I(\mathbf{X} + \mathbf{W}, t-1) = I(\mathbf{X}, t-1) + \nabla I(\mathbf{X}, t-1).\mathbf{W}$. This linearization yields the following motion constraint:

$$\hat{I}(\mathbf{X}, t) = I(\mathbf{X}, t-1) + \nabla I(\mathbf{X}, t-1).\mathbf{W} \tag{3}$$

3.4 Numerical resolution

Combining both constraints related, respectively to motion (3) and data model (1) leads to an equation model related to the displacement vector field:

$$\mathbb{M}(\mathbf{W})(\mathbf{X}, t) = I(\mathbf{X}, t-1) + \nabla I(\mathbf{X}, t-1).\mathbf{W} - I_G(\mathbf{X}, t) = 0 \tag{4}$$

On one hand, equation (4) is under determined as it is insufficiency to recover reliably the displacement vector field $\mathbf{W}$. On the other hand, a data model constraint involving a corrupted 3D representation, $I_G(\mathbf{X}, t)$, is embedded in this equation. To cope with the inaccuracy of $I_G(\mathbf{X}, t)$, we add an adapted smoothness term standing for spatial regularization of $\mathbf{W}$. This regularization overcomes the aforementioned ambiguity positioning problem. We estimate gradually a dense motion field using a variational formulation. We build a functional E whose minimum, with respect to $\mathbf{W}$, corresponds to the displacement vector field matching $I_G(\mathbf{X}, t)$ and $I(\mathbf{X}, t-1)$. E is composed of two additive terms:

$$E(\mathbf{W}) = \int_{\Omega} \mathbb{M}(\mathbf{W})^2(\mathbf{X}, t) d\mathbf{X} + \alpha \int_{\Omega} \Big(\|\nabla u\|^2 + \|\nabla v\|^2 + \|\nabla w\|^2 \Big) d\mathbf{X} \tag{5}$$

where the first term is related to equation (4) and the second one penalizes high spatial deformations of $\mathbf{W}$. Parameter α tunes the importance of the second term, *i.e.* the spatial regularization. Differentiation of E, with respect to displacements u, v and w yields a set of Euler-Lagrange equations (reader is referred to [6]). Discretization of the latter with finite differences leads to a huge linear system. We solve the minimization of the linear system within a Gauss-Seidel iterative scheme. Estimated 3D images are reconstructed by motion compensation with respect to equation (2). However, positions of mapped samples computed with this equation do not match perfectly grid-positions in the output image $\hat{I}(\mathbf{X}, t)$. We use then a method introduced in [7] to fit a smooth surface from the

set of mapped samples. This algorithm makes use of a coarse-to-fine hierarchy of control lattices in order to generate a sequence of bi-cubic B-spline functions whose sum approaches the desired interpolation function. In order to restore gradually the complete 3D+t sequence, we adopt a quite simple reconstruction strategy. We use, respectively, a forward progression procedure taking as first 3D available image, I_0 and a backward one using the final 3D available image I_1. Consequently, we generate two displacement vector fields at the median moment of the temporal interval $[t_0; t_1]$. The final 3D image is computed merely using the average of both fields.

4 Results

We present, in this section, a series of experiments in order to assess the performance of the proposed approach. They consist in computing the displacement vector field matching a 2D frame with an anterior 3D structure. As real data are not available, we design two sets of simple synthetic data. First one is composed of a couple of spheres $I(\mathbf{X}, t_1), I(\mathbf{X}, t_2)$ holding two squares lying on respectively the frontal and dorsal hemispheres. From t_1 to t_2, the frontal square center moves towards the right-bottom direction and the dorsal square evolves merely downwards. We simulate the 2D transformation of $I(\mathbf{X}, t_2)$ yielding $I_{2D}(\mathbf{x}, t)$. We build then inaccurate representation $I_G(\mathbf{X}, t_2)$ using $I_{2D}(\mathbf{x}, t)$ and finally we compute displacement vector field matching $I(\mathbf{X}, t_1)$ and $I_G(\mathbf{X}, t_2)$. The estimate $\hat{I}(\mathbf{X}, t_1)$ is computed by motion compensation. Second set of data is quite identical to the first one, except that squares occlude each other in the 2D projection. In order to visualize motion computation results, we use an adapted tool called **MAPVIS** [8]. This tool displays complex information (scalar and vectorial) lying on real spheroid surfaces or projected ones. In the 3D case, **MAPVIS** project data embedded on the actual globe observed part. Moreover, it improves visualization of texture in perspective projective views of spheric shaped structures. This tool is based on a suitable planar map projection that unrolls the curved surface around a given origin. Therefore, varying the projection origin around the surface allows to observe different views of the sphere. Since the selected map projection minimize distortions around the projection origin, the closer this point is the more accurate is the data recovery. Equatorial aspect of map projection, with reference to an origin lying on the equator displays the frontal hemisphere of the observed globe and polar one, with reference to the north geographical pole displays information lying on the northern hemisphere. We display in figures 3 and 4 motion results for respectively first and second set of synthetic data. Let us point out that motion computation produces a set of correct vector fields (from amplitude and direction point of views), perfectly tangent to the enclosing spherical surface. Moreover, our reconstruction method handle reliably ambiguity positioning as moving square are mapped on the right original hemispheres. It is noticeable by visualization of difference between the estimated 3D image and the original one, *i.e.* $I(., t_2) - \hat{I}(., t_2)$. Performance of our algorithm is not bothered by the more complex case showing object occlusion (see

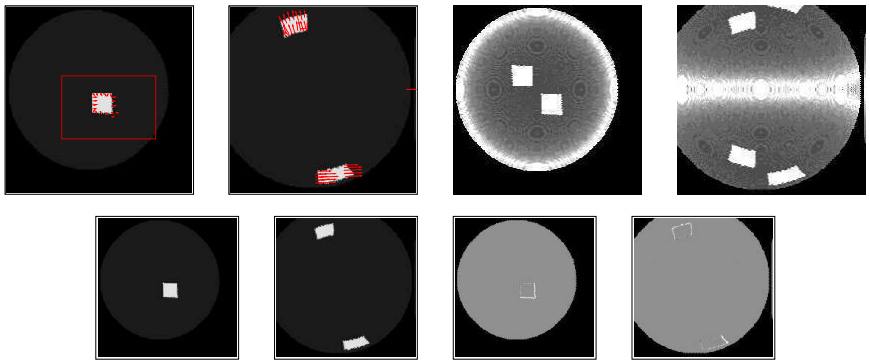


Fig. 3. On the first line: from left to right, visualization of the equatorial aspect then polar aspect of respectively of the set $I(.,t_1)$, $\mathbf{W}$ then $I_G(.,t_2)$. On the second line, from left to right, visualization of the equatorial aspect then polar aspect of respectively $\hat{I}(.,t_2)$ then of difference image $I(.,t_2) - \hat{I}(.,t_2)$.

figure 4) which does not ease the aforementioned structure distinction problem. However image difference show also reconstruction errors, doubtless amplified by the double interpolation procedure involved first to compute a smooth surface from samples recovered by motion compensation and second to visualize projective views [8]. These errors are localized in the border of moving objects. They may be caused by spatial derivative computations. Besides, they are due to the smoothing of motion discontinuities introduced by the spatial regularization of the estimated displacement vector field.

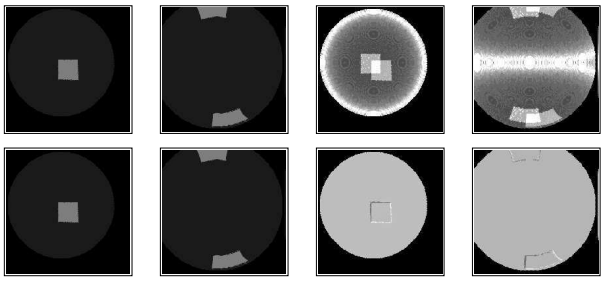


Fig. 4. On the first line: from left to right, visualization of the equatorial aspect then polar aspect of respectively $I(.,t_1)$ then $I_G(.,t_2)$. On the second line, from left to right, visualization of the equatorial aspect then polar aspect of respectively $\hat{I}(.,t_2)$ then of difference image $I(.,t_2) - \hat{I}(.,t_2)$.

5 Conclusion

In this paper, we have presented an original approach dedicated to 3D+t scene reconstruction using 3D temporal data and 2D+t temporal sequences. It is based on motion compensation and involves a prior geometrical data constraint. The latter touches spherical topology because several applications may be concerned. It leads us to build an inaccurate 3D inverse reconstruction of each 2D frame. This 3D image is enhanced by motion compensation, involving a regularity criterion under the brightness assumption constancy. The whole 3D+t sequence is reconstructed gradually using both 3D data as initial border conditions. Incorporating the prior geometrical constraint reduces relevantly the 3D-to-2D transformation to a merely a sum of contributions coming from the underlying frontal and dorsal hemispheres. This generates a positioning ambiguity on the intermediate 3D inaccurate representation. Our algorithm overcomes reliably this inaccuracy and handle possible object occlusions. Besides, we use an adapted tool, with relevant visualization properties, in order to display quickly reconstruction results. Currently, the whole sequence is computed frame per frame causing errors cumulation along the sequence. This limitation could be circumvented using a spatio-temporal smoothness constraint computed globally on the whole sequence [9]. Discussing other reconstruction issues, it is possible to generalize the prior geometrical constraint to any regular surface. We propose also in an other framework to solve the 3D+t scene recovery taken lower assumptions related to the data model. Moreover, we plan to apply our approach based on spherical topology to microscopic cell wall simulation acquisitions in the context described in [1]. That requires a spatial deconvolution framework to reconstruct each sphere from the original multi-focus image.

References

1. Staneva, G., Angelova, M., Koumanov, K.: Phospholipase a2 promotes raft budding and fission from giant liposomes. Chem Phys Lipids **129** (2004) 53–62
2. Natterer, F.: The mathematics of computerized tomography. Wiley, New York (1986)
3. Huei-Yung, L.: Computer vision techniques for complete 3D model reconstruction. PhD thesis, State university of New York at Stony Brook (2002)
4. Gueziec, A., Kazanzides, P., Williamson, B., Taylor, R.: Anatomy-based registration of CT-scan and intraoperative X-ray images for guiding a surgical robot. IEEE Trans Med Imaging **17** (1998) 715–728
5. Lemieux, L., Jagoe, R., Fish, D., Kitchen, N., Thomas, D.: a patient-tocomputed-tomography image registration method based on digitally reconstructed radiographs. Med Phys **21** (1994) 1749–1760
6. Glowinski, R.: Numerical Methods for Nonlinear Variational Problems. Springer edn. Series in computational physics, New York (1984)
7. Lee, S., Wolberg, G., Shin, S.: Scattered data interpolation with multilevel b-splines. IEEE Transactions on Visu and Comput Graph **3**(3) (1997) 228–244
8. Rekik, W., Béréziat, D., Dubuisson, S.: Mapvis: a map-projection based tool for visualizing scalar and vectorial information lying on spheroidal surfaces. In: Proceedings of IV05, London (July 2005)
9. Weickert, J., Schnrr, C.: Variational optic flow computation with a spatio-temporal smoothness constraint. Journal of Math Imag and Vision **14**(3) (May 2001) 245–255

A.4 Assimilation de données

[1] D. Béréziat and I. Herlin. Solving ill-posed image processing problems using data assimilation. *Numerical Algorithms*, to appear, 2010.

[2] D. Béréziat and I. Herlin. Using model of dynamics for large displacement estimation on noisy acquisitions. October 2010. Soumis à Transactions on Image Processing.

Numerical Algorithms manuscript No.
(will be inserted by the editor)

Solving ill-posed Image Processing problems using Data Assimilation

Dominique Béréziat · Isabelle Herlin

the date of receipt and acceptance should be inserted later

Abstract Data Assimilation is a mathematical framework used in environmental sciences to improve forecasts performed by meteorological, oceanographic or air quality simulation models. It aims to solve an evolution equation, describing the temporal dynamics, and an observation equation, linking the state vector and observations. In this article we use this framework to study a class of ill-posed Image Processing problems, usually solved by spatial and temporal regularization techniques. An approach is proposed to convert an ill-posed Image Processing problem in terms of a Data Assimilation system, solved by a 4D-Var method. This is illustrated by the estimation of optical flow from a noisy image sequence, with the dynamic model ensuring the temporal regularity of the result. The innovation of the paper concerns first, the extensive description of the tasks to be achieved for going from an image processing problem to a data assimilation description; second, the theoretical analysis of the covariance matrices involved in the algorithm; and third a specific discretisation scheme ensuring the stability of computation for the application on optical flow estimation.

Keywords computer vision, inverse problems, data assimilation, non linear advection, optical flow.

1 Introduction

In the research field of Image Processing, most problems are ill-posed (according to Hadamard definition) in the sense that it is not possible to provide a unique solution [4]. A first cause of ill-posedness is that the equations used to model image properties are under-determined. An example is given by the famous "aperture problem" occurring

D. Béréziat
Université Pierre et Marie Curie, LIP6, 4 place Jussieu, 75005 Paris, France.
E-mail: Dominique.Bereziat@upmc.fr
Tel.: +33 1 44 27 47 71 Fax.: +33 1 44 27 53 53

I. Herlin
INRIA, CEREA, Joint Laboratory ENPC-EDF R&D, Université Paris-Est,
Domaine de Voluceau, Rocquencourt - B.P. 105, 78153 Le Chesnay, France.

in the estimation of optical flow: a further constraint is required to compute a unique field of velocity vectors. As an image processing problem is usually modelled by a system of equations to be solved, the so-called *Image Model*, this type of ill-posedness means that the Image Model is not invertible. A second cause of ill-posedness occurs when the computation of image features can be obtained by different algorithms. For example, determining image gradient requires to approximate a differential operator by a discrete one among several possible finite difference formulations; each one with a different result.

An usual strategy to solve ill-posed problems is to provide additional information to the Image Model. Two options may be considered. 1) Explicit information: complementary images are used to enlarge the set of input data. However, this is generally not possible because additional acquisitions, having the requested properties, are not available. 2) Implicit information such as hypotheses on image properties or constraints on the solution. A usual way is to restrict the dimension of the space of admissible solutions. For instance, the result may be searched among the functions with bounded spatial variations, which is called "Tikhonov regularization method" in the literature [22]. In the general case, these additional properties or constraints are expressed as equations, which, combined to the Image Model, lead to a new invertible model.

Assuming not only one or two images but a whole sequence is available, enhancement may be obtained by taking into account the temporal evolution. Let us illustrate this on the image segmentation issue, which is a pure static problem. A spatial regularization method, such as Shah-Mumford's functional [10], produces a segmentation which is a compromise between a spatially smooth solution and the adequacy to the input data. If the segmentation process is performed directly on the whole sequence, following [26], the solution is then seek as a function depending on the spatial and temporal coordinates. This space-time approach has however several drawbacks. First, it imposes an arbitrary temporal regularity, which can not deal with complex dynamics. Second, missing data are taken into account in the process and introduce errors in the final solution. By "missing data", we refer to pixels' values displaying a wrong information due to either a failure of the acquisition system or noise. Third, as the solution is looked for in the space-time domain, this leads to an high computing complexity compared to a pure spatial model, which is a limiting factor for operational applications.

As the space-time approach fails in case of missing data, an alternative is to consider the temporal dynamics of the data. One challenge becomes to write an efficient dynamic model and to include it in the solution computation. Such information can be inferred, for instance, from *a priori* knowledge on the observed phenomena and related images. Moreover, data quality has to be evaluated in order to ignore missing data in the process. The first innovation of the paper is thus to use the dynamic model and the data quality measure to rewrite the image processing problem in a *Data Assimilation* system [8] in a generic way.

A Data Assimilation method solves a system of three equations with respect to a state vector, corresponding to the studied quantities:

- an evolution equation describes the evolution of the state vector over time, using an operator called the "evolution model";
- an "observation equation" models the links between the state vector and the observations provided by the image sequence;

– the initial condition of the state vector.

Each equation of the system only approximates the reality and a description of the error is stated in terms of a Gaussian noise characterized by a covariance matrix. For a given error, such matrix depicts the dependencies between error's coordinates on the one hand, and between two different locations in the space-time domain on the other hand.

We aim to use Data Assimilation as a generic framework to solve ill-posed Image Processing problems. The spatio-temporal regularization constraint is replaced by an evolution equation. Image dynamics being correctly approximated by this equation, the first drawback of the space-time approach is no more occurring. Describing the tasks to be achieved for going from an ill-posed image processing problem to a data assimilation system is the first contribution of this paper. A study of covariance matrices and their impact during the energy minimization has been performed. We show how a relevant choice allows to deal with missing data and solve the second drawback of the space-time approach. This is the second contribution of this work. These general considerations are illustrated on the problem of optical flow estimation. The third contribution is to propose a robust and innovative numerical scheme to implement the transport of velocity by itself, used in this paper as evolution equation. We show that the computation of the state vector can then be done locally in time, with specific choices for covariance matrices. This allows the algorithm to work on a frame-by-frame basis, unlike the space-time approach.

This article is organized as follows. Section 2 introduces the concept and difficulties of ill-posed problems in the Image Processing research field. We give some typical examples and present a short start-of-the-art of Tikhonov regularization methods. Section 3 describes the variational Data Assimilation method known as the 4D-Var algorithm. How can Data Assimilation be used to solve ill-posed problems by assimilating images in an appropriate evolution model is explained in Section 4. Section 5 is then a direct application describing how to compute optical flow in this framework, using a robust numerical scheme. It also presents and discusses experimental results. We conclude in Section 6 and give some scientific perspectives to this research study.

2 Ill-posed problems in Image Processing

Hadamard gave the following definition: a problem is well-posed if 1) it has a unique solution, 2) the solution depends continuously on the input [4]. A problem which does not meet these conditions is called ill-posed. Using this definition, optical flow estimation, image registration, curves or surfaces matching, tracking of multiple objects, segmentation, restoration, deconvolution, denoising and shape from shading are well known ill-posed image processing problems because the equations used for modeling them are under-constrained (see the aperture problem of optical flow for instance).

Links between image properties and the solution are modeled as a set of equations constituting the so-called Image Model. The image, input data, is denoted $\mathbf{Y}$ and depends on the spatial coordinate $\mathbf{x}$ in a bounded domain denoted Ω. The solution of the problem, denoted $\mathbf{X}$, is not necessarily an image: it can be a velocity field, a curve, *etc.* To be general, the Image Model is mathematically written as:

$$\mathbb{I}(\mathbf{X}, \mathbf{Y})(\mathbf{x}) = \mathbf{0} \quad \forall \mathbf{x} \in \Omega \tag{1}$$

with $\mathbb{I}$ a differentiable operator that may be:

- linear: $\mathbb{I}(\mathbf{X},\mathbf{Y}) = \mathbf{Y} - A(\mathbf{X})$ with A linear. This is a typically the case for segmentation, restoration, denoising, deconvolution: $\mathbb{I}$ measures the discrepancy between the input image and the solution filtered by the operator A.
- non linear. A common situation in image processing is the following: $\mathbb{I}(\mathbf{X},\mathbf{Y}) = B(\mathbf{Y}(C(\mathbf{X}))$. The input data $\mathbf{Y}$ is only considered on the pixels $C(\mathbf{X})$ depending on the solution. For example, $\mathbb{I}(\mathbf{X},\mathbf{Y})(\mathbf{x}) = -\|\nabla\mathbf{Y}(\mathbf{X}(\mathbf{x}))\|$ for the well known active contour or $\mathbb{I}(\mathbf{X},Y_1,Y_2)(\mathbf{x}) = Y_2(\mathbf{x}+\mathbf{X}(\mathbf{x})) - Y_1(\mathbf{x})$ for estimating the optical flow $\mathbf{X}$ between two images Y_1 and Y_2.

In this section, attention is focused on variational methods: instead of directly solving (1), an optimization problem is formulated and the solution is obtained by minimizing $E(\mathbf{X}) = \int_\Omega \Psi(\|\mathbb{I}(\mathbf{X},\mathbf{Y})\|)d\mathbf{x}$ with Ψ a convex function such as $\Psi(0) = 0$. As Equation (1) is under-constrained, a possible method to obtain a unique solution is to use the Tikhonov regularization. This is performed by adding a second term to the functional E which becomes:

$$E(\mathbf{X}) = \int_\Omega \left(\Psi(\|\mathbb{I}(\mathbf{X},\mathbf{Y})\|) + \sum_{n\geq 0} \alpha_n \Psi\left(\left\| \frac{\partial^n \mathbf{X}}{\partial \mathbf{x}^n} \right\| \right) \right) d\mathbf{x} \tag{2}$$

An usual choice is to set $\alpha_n = 0$ for $n \neq 1$ ensuring a first order regularization or $\alpha_n = 0$ for $n \neq 1, 2$ ensuring a second order regularization. The norm used in the regularizing term is often the Euclidean norm but other choices are possible: for instance, to determine optical flow, Nagel uses in [11] an oriented norm driven by the local image configuration. The regularization performed by the algorithm is then weak on edges and strong otherwise. The minimization of E is led in the calculus of variation framework: the solution is searched as the zero of the Euler-Lagrange equation associated to (2): $\frac{\partial E}{\partial \mathbf{X}} = 0$, with $\frac{\partial E}{\partial \mathbf{X}}$ denoting the differential of E with respect to $\mathbf{X}$. With a first order regularization, $\frac{\partial}{\partial \mathbf{x}}$ is usually denoted ∇, and
$E(\mathbf{X}) = \int \left(\Psi(\|\mathbb{I}(\mathbf{X},\mathbf{Y})\|) + \alpha_1 \Psi(\|\nabla\mathbf{X}\|)\right) d\mathbf{x}$. The general expression of the corresponding Euler-Lagrange equation becomes:

$$\Psi'(\|\mathbb{I}(\mathbf{X},\mathbf{Y})\|)\frac{\partial \mathbb{I}}{\partial \mathbf{X}} - \alpha_1 \nabla.\left(\Psi'(\|\nabla\mathbf{X}\|)\frac{\nabla\mathbf{X}}{\|\nabla\mathbf{X}\|} \right) = 0 \tag{3}$$

It is discretized by finite differences and the solution is obtained using a Jacobi or Gauss-Seidel method in the linear case and a method of steepest descent or conjugate gradient otherwise.

From the beginning of this section, the image $\mathbf{Y}$ is considered as only depending on the spatial coordinate $\mathbf{x}$ and the Image Model $\mathbb{I}$ is then static. In the case of optical flow computation, at least two frames are required to compute the temporal derivatives, but the result obtained on one image, using (3), has no link with those obtained on adjacent frames. An improved solution, when dealing with a temporal sequence, has been proposed by Weickert *et al* [26] and consists in minimizing the functional:

$$E(\mathbf{X}) = \int_\Omega \int_0^{\mathbf{T}} \left(\Psi(\|\mathbb{I}(\mathbf{X},\mathbf{Y})\|) + \sum_{n\geq 0} \alpha_n \Psi\left(\left\| \frac{\partial^n \mathbf{X}}{\partial \mathbf{x}^n} \right\|, \left\| \frac{\partial \mathbf{X}}{\partial t} \right\| \right) \right) d\mathbf{x}dt \tag{4}$$

with $\mathbf{X}$ and $\mathbf{Y}$ becoming space-time dependent. Such a functional describes accurately linear dynamics but becomes irrelevant otherwise. Moreover, missing data are taken

into account during the computation: the method performs a smoothing of these aberrant values on the spatio-temporal neighborhood. Last, the time being viewed as an additional dimension, a consequence is the proportional increase of the problem's size. A huge memory is then requested for processing a large sequence.

A partial answer to the problem of dealing with complex dynamics and missing data is to include a model of the temporal evolution. This implies to be able to solve simultaneously the Image Model $\mathbb{I}$ and the evolution equation. This possibility is offered by the Data Assimilation framework described in the following Section.

3 The Data Assimilation framework

3.1 Mathematical setting

Let $\mathbf{X}$ being the state vector depending on the spatial coordinate $\mathbf{x}$ and time t. $\mathbf{X}$ is defined on $A = \Omega \times [0, T]$, Ω being the spatial domain and $[0, \mathbf{T}]$ the temporal domain.

We assume $\mathbf{X}$ is evolving in time according to:

$$\frac{\partial \mathbf{X}}{\partial t}(\mathbf{x}, t) + \mathbb{M}(\mathbf{X})(\mathbf{x}, t) = \mathcal{E}_m(\mathbf{x}, t) \tag{5}$$

$\mathbb{M}$, named *evolution model*, is supposed differentiable. As $\mathbb{M}$ may describe approximately the evolution of the state vector, a *model error* $\mathcal{E}_m$ is introduced to quantify the imperfections.

We consider having knowledge on the initial condition of the state vector at $t = 0$:

$$\mathbf{X}(\mathbf{x}, 0) = \mathbf{X}_b(\mathbf{x}) + \mathcal{E}_b(\mathbf{x}) \tag{6}$$

with $\mathbf{X}_b$ named the *background value* and $\mathcal{E}_b$ denotes the *background error*.

Observations $\mathbf{Y}(\mathbf{x}, t)$ are available at location $\mathbf{x}$ and date t and linked to the state vector through the observation equation:

$$\mathbf{Y}(\mathbf{x}, t) = \mathbb{H}(\mathbf{X})(\mathbf{x}, t) + \mathcal{E}_O(\mathbf{x}, t) \tag{7}$$

Equation (7) is the standard form of the observation equation used in the Data Assimilation literature. However, this formulation is quite restrictive to describe the links, possibly complex, existing between the observations and the state vector. To be more general, the following will be used in this paper:

$$\mathbb{H}(\mathbf{Y}, \mathbf{X})(\mathbf{x}, t) = \mathcal{E}_O(\mathbf{x}, t) \tag{8}$$

which includes the previous formulation (7). The *observation error* $\mathcal{E}_O$ simultaneously represents the imperfection of the observation operator $\mathbb{H}$ and the measurement errors.

$\mathcal{E}_m$, $\mathcal{E}_b$ and $\mathcal{E}_O$ are assumed to be Gaussian and then fully characterized by their covariance matrices Q, B and R [21]. Let $\mathbf{Z}$ denote a Gaussian stochastic vector depending on a space-time coordinate $(\mathbf{x}, t)$; $\mathbf{Z} = \mathbf{Z}(x, t)$ and $\mathbf{Z}' = \mathbf{Z}(x', t')$ are random vectors on the two given locations. The covariance matrix Σ, computed for $\mathbf{Z}$ and $\mathbf{Z}'$, measures their dependency and is defined by:

$$\Sigma(\mathbf{x}, t, \mathbf{x}', t') = \int (\mathbf{Z} - \mathbb{E}\mathbf{Z})^T (\mathbf{Z}' - \mathbb{E}\mathbf{Z}') dP_{\mathbf{Z},\mathbf{Z}'} \tag{9}$$

with $P_{\mathbf{Z},\mathbf{Z}'}$ the joint distribution of $\left(\mathbf{Z}, \mathbf{Z}'\right)$ and $\mathbb{E}$ denoting the expectation.

3.2 Variational formulation

In order to solve the system (5), (6) and (8) with respect to $\mathbf{X}$ having a maximal probability, the following functional, to be minimized, is defined:

$$\begin{aligned} E(\mathbf{X}) =& \frac{1}{2}\int_A\int_A \left(\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X})\right)^T (\mathbf{x},t) Q^{-1}(\mathbf{x},t,\mathbf{x}',t') \left(\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X})\right)(\mathbf{x}',t') d\mathbf{x}dtd\mathbf{x}'dt' \\ &+ \int_A\int_A \mathbb{H}(\mathbf{X},\mathbf{Y})^T(\mathbf{x},t) R^{-1}(\mathbf{x},t,\mathbf{x}',t') \mathbb{H}(\mathbf{X},\mathbf{Y})(\mathbf{x}',t') d\mathbf{x}dtd\mathbf{x}'dt' \\ &+ \int_\Omega\int_\Omega \big(\mathbf{X}(\mathbf{x},0) - \mathbf{X}_b(\mathbf{x})\big)^T B^{-1}(\mathbf{x},\mathbf{x}') \big(\mathbf{X}(\mathbf{x}',0) - \mathbf{X}_b(\mathbf{x}')\big) d\mathbf{x}d\mathbf{x}' \end{aligned} \tag{10}$$

As $\mathcal{E}_m$, $\mathcal{E}_b$ and $\mathcal{E}_O$ are assumed to be independent, the functional E represents the log-density of $\mathbf{X}$ law [2]. The minimization is carried out by solving the associated Euler-Lagrange equation. The differential $\dfrac{\partial E}{\partial \mathbf{X}}$ is obtained by computing the derivative of E with respect to $\mathbf{X}$ in direction η:

$$\frac{\partial E}{\partial \mathbf{X}}(\eta) = \lim_{\gamma\to 0} \frac{d}{d\gamma}(E(\mathbf{X}+\gamma\eta)) \tag{11}$$

and by introducing an auxiliary variable λ, called the *adjoint variable* in the literature of Data Assimilation:

$$\lambda(\mathbf{x},t) = \int_A Q^{-1}(\mathbf{x},t,\mathbf{x}',t') \left(\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X})\right)(\mathbf{x}',t') d\mathbf{x}'dt' \tag{12}$$

We detail in Appendix A the determination of the Euler-Lagrange equation associated to (10). This leads to the following so-called Optimality System [9]:

$$\lambda(\mathbf{x},\mathbf{T}) = 0 \tag{13}$$

$$-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^* \lambda = -\int_A \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* (\mathbf{x},t) R^{-1} \mathbb{H}(\mathbf{X},\mathbf{Y})(\mathbf{x}',t') d\mathbf{x}'dt' \tag{14}$$

$$\mathbf{X}(\mathbf{x},0) = \int_\Omega B\lambda(\mathbf{x}',0) d\mathbf{x}' + \mathbf{X}_b(\mathbf{x}) \tag{15}$$

$$\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}) = \int_A Q\lambda(\mathbf{x}',t') d\mathbf{x}'dt' \tag{16}$$

Because the initial condition for λ is given at time T (13), λ must be computed backward in time using (14). Equation (14) makes use of two *adjoint operators* denoted $\left(\dfrac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^*$ and $\left(\dfrac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^*$. Adjoint operators are also called dual operators in the mathematics literature. For a given operator $\mathbb{K}$, we have:

$$\int (\mathbb{K}(\eta))^T \lambda d\mu = \int \eta^T \mathbb{K}^*(\lambda) d\mu \tag{17}$$

for all integrable functions η and λ. Riesz's theorem ensures the existence and uniqueness of the adjoint operator. For clarifying the discussion, let us determine the adjoint

operator of $\mathbb{K} = \dfrac{\partial}{\partial x}$ in an interval $[a, b]$:

$$\begin{aligned}\int_a^b \frac{\partial f}{\partial x}(x)g(x)dx &= [f(x)g(x)]_{x=a}^{x=b} - \int_a^b f(x)\frac{\partial g}{\partial x}(x)dx \\ &= \int_a^b f(x)\left(\left(\delta(x-b) - \delta(x-a)\right)g(x) - \frac{\partial g}{\partial x}(x)\right)dx \\ &= \int_a^b f(x)\left(\frac{\partial g}{\partial x}\right)^*(x)dx\end{aligned}$$

The adjoint operator of $\dfrac{\partial}{\partial x}$ in $[a, b]$ is then $\left(\dfrac{\partial}{\partial x}\right)^* = \delta(x-b) - \delta(x-a) - \dfrac{\partial}{\partial x}$.
If $\mathbb{K}$ is a differential operator, the adjoint operator is then a compact notation for integration by parts.

3.3 Incremental algorithm

Solving the Optimality System is however not straightforward: the state vector is determined by equations (15,16) using the adjoint variable and the adjoint variable is determined by equations (13,14) using the state vector. To break this deadlock, an incremental method is applied. The underlying idea comes from the following lemma:

$$\min_{w \in \mathcal{V}(w_0)} E(w) = \min_{\delta w \in \mathcal{V}(0)} E(w_0 + \delta w)$$

where w_0 denotes a local minimum of E and $\mathcal{V}(w_0)$ denotes one neighborhood of w_0. The state vector is therefore written as $\mathbf{X}_b + \delta\mathbf{X}$ where $\mathbf{X}_b$ is called the *background variable* in the Data Assimilation literature, and $\delta\mathbf{X}$ is the incremental variable. $\mathbf{X}$ is then replaced by $\mathbf{X}_b + \delta\mathbf{X}$ in equations (14), (15) and (16). If $\mathbb{M}$ and $\mathbb{H}$ are non linear operators, we obtain:

$$\mathbb{M}(\mathbf{X}) \simeq \mathbb{M}(\mathbf{X}_b + \delta\mathbf{X}) = \mathbb{M}(\mathbf{X}_b) + \frac{\partial \mathbb{M}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b}(\delta\mathbf{X}) \tag{18}$$

$$\mathbb{H}(\mathbf{X}, \mathbf{Y}) \simeq \mathbb{H}(\mathbf{X}_b + \delta\mathbf{X}, \mathbf{Y}) = \mathbb{H}(\mathbf{X}_b, \mathbf{Y}) + \frac{\partial \mathbb{H}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b}(\delta\mathbf{X}) \tag{19}$$

from a first order Taylor development of $\mathbb{M}$ and $\mathbb{H}$ at $\mathbf{X}_b$. Equations (18) and (19) become equalities if $\mathbb{M}$ and $\mathbb{H}$ are linear. In both cases, (18,19) lead to the following new system:

$$\lambda(\mathbf{x}, \mathbf{T}) = 0 \tag{20}$$

$$-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b}\right)^* \lambda = -\int_A \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b}\right)^* R^{-1}\left(\mathbb{H}(\mathbf{X}_b, \mathbf{Y}) + \frac{\partial \mathbb{H}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b}(\delta\mathbf{X})\right) d\mathbf{x}'dt' \tag{21}$$

$$\mathbf{X}_b(\mathbf{x}, 0) = \mathbf{X}_b(\mathbf{x}) \tag{22}$$

$$\frac{\partial \mathbf{X}_b}{\partial t} + \mathbb{M}(\mathbf{X}_b) = 0 \tag{23}$$

$$\delta\mathbf{X}(\mathbf{x}, 0) = \int_\Omega B\lambda(\mathbf{x}', 0)d\mathbf{x}' \tag{24}$$

$$\frac{\partial \delta\mathbf{X}}{\partial t} + \frac{\partial \mathbb{M}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b}(\delta\mathbf{X}) = \int_A Q\lambda(\mathbf{x}', t')d\mathbf{x}'dt' \tag{25}$$

The background variable $\mathbf{X}_b$ is first calculated from equations (22) and (23). The adjoint variable λ is then obtained from the background variable using equations (20) and (21). Last, the incremental variable $\delta\mathbf{X}$ is obtained from the adjoint variable using equations (24) and (25).

If $\mathbb{M}$ and $\mathbb{H}$ are not linear, equations (20,21,24,25) only produce an approximated solution $\mathbf{X}_b + \delta\mathbf{X}$ due to the first order Taylor development described in (18) and (19). In this case, the incremental algorithm is applied iteratively until convergence. This iterative method, indexed by i, is summarized in the following:

1. Initialization ($i = 0$) :
   (a) Compute the background variable $\mathbf{X}_b$ from the initial condition of the state vector at $t = 0$ in (6):

$$\mathbf{X}_b(\mathbf{x}, 0) = \mathbf{X}_b(\mathbf{x})$$
$$\frac{\partial \mathbf{X}_b}{\partial t} + \mathbb{M}(\mathbf{X}_b) = \mathbf{0}, \forall t \in [0, \mathbf{T}]$$

   (b) Initialize the incremental variable:

$$\delta\mathbf{X}(\mathbf{x}, t) = \mathbf{0}, \forall t \in [0, \mathbf{T}]$$

2. Do $i = i + 1$ and :
   (a) Compute the adjoint variable λ from $t = T$ to $t = 0$:

$$\lambda(\mathbf{x}, T) = \mathbf{0}$$
$$-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b}\right)^* (\lambda) = -\int \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b, \mathbf{Y}}\right)^* R^{-1} \left[\mathbb{H}(\mathbf{X}_b, \mathbf{Y}) + \frac{\partial \mathbb{H}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b, \mathbf{Y}} (\delta\mathbf{X})\right] d\mathbf{x}' dt'$$

   (b) Update the value of background variable:

$$\mathbf{X}_b(\mathbf{x}, t) = \mathbf{X}_b(\mathbf{x}, t) + \delta\mathbf{X}(\mathbf{x}, t)$$

   (c) Compute the incremental variable $\delta\mathbf{X}$ (required for the next iteration) from $t = 0$ to $t = T$:

$$\delta\mathbf{X}(\mathbf{x}, 0) = \int B(\mathbf{x}, \mathbf{x}')\lambda(\mathbf{x}', 0) d\mathbf{x}'$$
$$\frac{\partial \delta\mathbf{X}}{\partial t} + \frac{\partial \mathbb{M}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b} (\delta\mathbf{X}) = \int Q(\mathbf{x}, t, \mathbf{x}', t')\lambda(\mathbf{x}', t') d\mathbf{x}' dt'$$

   Back to step 2 until $\|\delta\mathbf{X}\|^2 \leq \epsilon$.
3. Final result is $\mathbf{X}_b + \delta\mathbf{X}$.

4 Assimilation of images

This section explains how to solve the ill-posed Image Processing problems using the framework of Data Assimilation and constitutes the core of this research and one main contribution of the paper. Using Data Assimilation to solve Image Processing problems is a relatively recent domain. This idea has been described in [5] and the first application has been published in [17]. Studies have been done on curve tracking [16, 17] and

determination of optical flow [14,15]. In [14] two evolution models are considered, a first one describing fluid motion by the transport of vorticity and a second one describing video motion using the transport of velocity by itself, stabilized by a diffusion process. In [15] a shallow water model is considered to describe the atmospheric circulation. In [7], a method is proposed to estimate the ocean surface circulation from SST data: images are assimilated in a dynamic image model to produce pseudo-observations for the oceanographic model. In this paper, we restrict to the case of ill-posed problems, which have been solved in the literature using Tikhonov regularization methods. We define a method to convert these problems, in a generic way, from the Tikhonov regularization to the Data Assimilation framework. The proposed method, which is the first innovation of the paper, rewrites the Image Model and the regularization term under the form of the system of three equations (5,6,8).

- First, the state and observation vectors have to be defined. Obviously, the observations will be images or processed images, but the components of the state vector will strongly depend on the studied problem. For example, segmentation, denoising and restoration use a state vector which is composed of the result's values for all pixels of the input image. Tracking, image registration and motion estimation use a state vector whose components are the values of the resulting vector field. Active contours use a curve.
- Second, a suitable equation describing the temporal evolution of the state vector has to be stated (Subsection 4.2). Next, an observation equation is written expressing the links between the state vector and the images (Subsection 4.3). Sometimes, this link is highly indirect. For instance, as it is not possible to deduce the ocean circulation from surface temperatures with a shallow-water model, a specific system has to be stated with relevant evolution and observation equations [7].
- Third, the initial condition should be defined.
- Last, the errors, characterized by their covariance matrices, must be chosen for the three equations. It is therefore crucial to understand the role of these matrices inside functional (10). The second innovation of the paper is therefore to conduct a full analysis of the errors and associated covariance matrices and their impact in term of spatio-temporal regularization (Subsection 4.1).

4.1 Covariance matrix

A covariance matrix Σ being defined by equation (9), its inverse is formally and implicitly defined [13] as:

$$\int \Sigma^{-1}(\mathbf{x},\mathbf{x}'')\Sigma(\mathbf{x}'',\mathbf{x}')d\mathbf{x}'' = \delta(\mathbf{x}-\mathbf{x}') \tag{26}$$

We consider three possible choices of covariance and analyze their respective impact in a functional $\iint F^T(\mathbf{Z})\Sigma^{-1}F(\mathbf{Z})d\mathbf{x}d\mathbf{x}'$ which has to be minimized.

As a first example, let Σ be the Dirac covariance defined by $\Sigma(\mathbf{x},\mathbf{x}') = \delta\left(\mathbf{x}-\mathbf{x}'\right)$. This covariance expresses a null interaction between two space locations. The Dirac function, $\delta\left(\right)$, has the following property:

$$\int_{\Omega} \delta\left(x'\right)\delta\left(x-x'\right)dx' = \delta\left(x\right) \tag{27}$$

By identification of (26) and (27), we have $\Sigma^{-1}(\mathbf{x},\mathbf{x}') = \delta\left(\mathbf{x}-\mathbf{x}'\right)$ and:

$$\begin{aligned} &\iint_{\Omega^2} F(\mathbf{Z})^T(\mathbf{x})\Sigma^{-1}(\mathbf{x},\mathbf{x}')F(\mathbf{Z})(\mathbf{x}')d\mathbf{x}d\mathbf{x}' \\ &= \int_{\Omega} F(\mathbf{Z})^T(\mathbf{x})F(\mathbf{Z})(\mathbf{x})d\mathbf{x} = \int_{\Omega} \|F(\mathbf{Z})\|^2 d\mathbf{x} \end{aligned} \tag{28}$$

A Dirac covariance is therefore acting like a zero-order regularization on $F(\mathbf{Z})$.

More generally, we consider the case of an isotropic covariance which is written as $\Sigma(\mathbf{x},\mathbf{x}') = s(\mathbf{x}-\mathbf{x}')$. In this case, determining its inverse from (26) is equivalent to determine its inverse convolution defined by:

$$s^{-1} \star s(\mathbf{x}) = \delta(\mathbf{x}) \tag{29}$$

This is more easily done in the Fourier domain as the latter equation is equivalent to:

$$\widehat{s^{-1} \star s}(\omega) = 1$$

Using the convolution theorem, we have:

$$\begin{aligned} \widehat{s^{-1}}(\omega)\widehat{s}(\omega) &= 1 \\ \widehat{s^{-1}}(\omega) &= \frac{1}{\widehat{s}(\omega)} \end{aligned}$$

and the inverse convolution $s^{-1}(x)$ is obtained by the inverse Fourier transform of $\frac{1}{\widehat{s}}$.

Let us apply this with the exponential covariance defined by $s(\mathbf{x}) = \exp\left(-\frac{|\mathbf{x}|}{\sigma}\right)$. Its Fourier transform is $\frac{2\sigma}{1+\sigma^2\omega^2}$. We have:

$$\begin{aligned} \widehat{s^{-1}}(\omega) &= \frac{1+\sigma^2\omega^2}{2\sigma} \\ s^{-1}(\mathbf{x}) &= \frac{1}{2\sigma}\left(\delta(\mathbf{x}) - \sigma^2\delta''(\mathbf{x})\right) \end{aligned}$$

We replace the expression of $\Sigma^{-1}(\mathbf{x},\mathbf{x}')$ by $s^{-1}(\mathbf{x}-\mathbf{x}')$ in the functional (28):

$$\begin{aligned} &\iint_{\Omega^2} F(\mathbf{Z})^T(\mathbf{x})\Sigma^{-1}(\mathbf{x},\mathbf{x}')F(\mathbf{Z})(\mathbf{x}')d\mathbf{x}d\mathbf{x}' \\ &= \frac{1}{2\sigma}\int_{\Omega} F(\mathbf{Z})^T(\mathbf{x})\left(F(\mathbf{Z}) - \sigma^2\frac{\partial^2 F(\mathbf{Z})}{\partial\mathbf{x}^2}\right)d\mathbf{x} \\ &= \frac{1}{2\sigma}\int_{\Omega}\left(\|F(\mathbf{Z})\|^2 + \sigma^2\left\|\frac{\partial F(\mathbf{Z})}{\partial\mathbf{x}}\right\|^2\right)d\mathbf{x} \end{aligned} \tag{30}$$

Integration by parts is applied for deriving (30), assuming null boundary terms. The exponential covariance is thus associated to a first-order regularization of $F(\mathbf{Z})$.

Another possibility is the Gaussian covariance defined by $s(\mathbf{x}) = \exp\left(-\frac{\mathbf{x}^2}{\sigma^2}\right)$. Using the previous technique, the Fourier transform of the Gaussian function is first established: $\widehat{s}(\omega) = \sigma\sqrt{\pi}\exp(-\frac{\sigma^2}{4}\omega^2)$. The Fourier transform of the inverse covariance

is thus $\widehat{s^{-1}}(\omega) = \frac{1}{\sigma\sqrt{\pi}} \exp\left(\frac{\sigma^2}{4}\omega^2\right)$, but its inverse Fourier transform can not be directly computed. Let us replace the exponential by its infinite series:

$$\begin{aligned}\widehat{s^{-1}}(\omega) &= \frac{1}{\sigma\sqrt{\pi}} \sum_{n\geq 0} \frac{1}{n!} \left(\frac{\sigma^2}{4}\omega^2\right)^n \\ &= \frac{1}{\sigma\sqrt{\pi}} \sum_{n\geq 0} a_n \omega^{2n} \end{aligned} \tag{31}$$

with $a_n = \frac{1}{n!}\left(\frac{\sigma}{2}\right)^{2n}$. Remembering that 1 is the inverse Fourier transform of the Dirac function and using the theorem of Fourier derivation, equation (31) leads to:

$$\begin{aligned} s^{-1}(\mathbf{x}) &= \frac{1}{\sigma\sqrt{\pi}} \sum_{n\geq 0} a_n (i)^{2n} \frac{\partial^{2n}}{\partial x^{2n}} \delta(\mathbf{x}) \\ &= \frac{1}{\sigma\sqrt{\pi}} \sum_{n\geq 0} a_n (-1)^n \delta^{(2n)}(\mathbf{x}) \end{aligned}$$

Let us examine the impact of such covariance in the functional (28):

$$\begin{aligned} &\iint F(\mathbf{Z})^T(\mathbf{x}) \Sigma^{-1}(\mathbf{x},\mathbf{x}') F(\mathbf{Z})(\mathbf{x}') d\mathbf{x} d\mathbf{x}' \\ &= \frac{1}{\sigma\sqrt{\pi}} \sum_{n\geq 0} (-1)^n a_n \int F(\mathbf{Z})^T(\mathbf{x}) \left(\int \delta^{(2n)}(\mathbf{x}-\mathbf{x}') F(\mathbf{Z})(\mathbf{x}') d\mathbf{x}' \right) d\mathbf{x} \\ &= \frac{1}{\sigma\sqrt{\pi}} \sum_{n\geq 0} (-1)^n a_n \int F(\mathbf{Z})^T(\mathbf{x}) F^{(2n)}(\mathbf{Z})(\mathbf{x}) d\mathbf{x} \\ &= \frac{1}{\sigma\sqrt{\pi}} \int \sum_{n\geq 0} a_n \left\| \frac{\partial^n F(\mathbf{Z})}{\partial \mathbf{x}^n} \right\|^2 (\mathbf{x}) d\mathbf{x} \end{aligned}$$

Again, assuming null boundary conditions, we use a series of integration by parts to derive to the final result. The Gaussian exponential covariance then corresponds to the Tikhonov regularization at any order of $F(\mathbf{Z})$.
The inversion of a covariance matrix Σ is however non-trivial and usually inaccessible. Restrictive choices have to be made such as those previously described. In the general case, it still remains possible to approximate the matrix of covariance using finite difference operators and to inverse it using numerical techniques. Unfortunately, if the discrete matrix is large, the inversion is costly and often numerically unstable. For further details, the reader is referred to [13,21].

4.2 The evolution model

Tikhonov regularization is in fact assuming spatial properties of the result from heuristics on the dynamics. These heuristics, with a better knowledge on the image dynamics, could be efficiently written as an evolution equation of the state vector $\mathbf{X}$. This evolution law imposes a temporal regularity of $\mathbf{X}$ and consequently impacts in specific spatial configurations.

A first and simple heuristic about dynamics is to assume $\mathbf{X}$ being constant over time, which is expressed by:

$$\frac{d\mathbf{X}}{dt} = 0$$

or:

$$\frac{\partial \mathbf{X}}{\partial t} + \frac{\partial \mathbf{X}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial t} = 0 \tag{32}$$

This is a transport equation as $\frac{\partial \mathbf{x}}{\partial t}$ is a velocity vector. Identifying equation (32) and (5), the evolution model is $\mathbb{M}(\mathbf{X})(\mathbf{x}, t) = \frac{\partial \mathbf{X}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial t}$. An example of using (32) as evolution equation is given in Section 5 for optical flow estimation.

Another possible heuristic is to express the transport of the state vector as a diffusion process, a physical law applying for chemical species or temperature. The general formulation is: $\frac{\partial \mathbf{X}}{\partial t} = \nabla^T (D \nabla \mathbf{X})$ and by identification with equation (5), the evolution model is $\mathbb{M}(\mathbf{X}) = -\nabla^T (D \nabla \mathbf{X})$. The matrix D is a tensor characterizing simultaneously the direction and the intensity of the diffusion. If D does not depend on spatial coordinates, the diffusion is linear and equivalent to a smoothing process using a Gaussian convolution [27]. It is also possible to drive the diffusion according to image characteristics. A standard example is the Perona & Malik diffusion [18]: the tensor matrix D is equal to $c(\|\nabla \mathbf{X}\|)Id$ with c a Gaussian function and Id the identity matrix, with the result of smoothing the image on homogeneous regions and preserving contours. The tensor matrix D can also take into account the orientation of the image gradient [24, 25] resulting in spatial regularization properties similar to that of Nagel [11].

The two previous definitions of $\mathbb{M}$ remain restrictive for image sequences displaying complex dynamics. In these cases, the evolution model could be built using prior information on the observed data. As a first example, dynamics can be approximated by piecewise linear functions whose parameters are estimated by analyzing the evolution of the image probability density function with a particule filter method. However, this issue remains complex and widely open. As a second example, an evolution model based on a physical law is considered. In [7,15] a shallow-water equation is chosen for modeling the evolution of apparent motion, which is an advection-diffusion equation with additional forcing terms. However, this is specific to the ocean surface circulation and can not be applied to a generic framework.

The covariance matrix Q, associated to the model error $\mathcal{E}_m$, is used in functional (10) for regularizing $\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X})$. For this reason Q is chosen as Dirac covariance as explained in Subsection 4.1. In such a way, we are ensuring that the temporal evolution of $\mathbf{X}$ is coherent with the heuristics involved in the model $\mathbb{M}$.

4.3 The observation equation

As previously pointed out, the observation equation describes the links between the state vector and the observations. In the standard framework of Image Processing, an image or a sequence of images provides the observations in the Image Model $\mathbb{I}$ expressed in equation (1). The operator $\mathbb{H}$, as it appears in equation (8), is then defined as the image operator $\mathbb{I}$ i.e. $\mathbb{H}(\mathbf{X}, \mathbf{Y}) \equiv \mathbb{I}(\mathbf{X}, \mathbf{Y})$.

The observation error $\mathcal{E}_{\mathbf{Y}}$, characterized by its covariance matrix R, has also to be specified. R weights the contribution of observations in equation (21). Its inverse

should then have values close to zero when observations should be discarded, which is the case for missing data. Using the Dirac covariance, R is then written as:

$$R(\mathbf{x}, t, \mathbf{x}', t') = r(\mathbf{x}, t)\delta\left(\mathbf{x} - \mathbf{x}'\right)\delta\left(t - t'\right)$$

with r a real matrix whose size corresponds to the number of components of $\mathbb{H}$ and the inverse is:

$$R^{-1}(\mathbf{x}, t, \mathbf{x}', t') = \delta\left(\mathbf{x} - \mathbf{x}'\right)\delta\left(t - t'\right) r^{-1}(\mathbf{x}, t) \tag{33}$$

The matrix r^{-1} characterizes the quality of the observation: a high value indicates that the observation value is relevant and a value close to zero indicates an irrelevant observation value, which should not be included in the computation of the solution. Assuming the availability of a function f measuring the confidence in observation data ($f \in [0, 1]$, $f = 0$ for no confidence), one possible formulation of r^{-1} is:

$$r^{-1}(\mathbf{x}, t) = r_0(1 - f(\mathbf{x}, t)) + r_1 f(\mathbf{x}, t) \tag{34}$$

$r^{-1}(\mathbf{x}, t)$ will be equal to a "minimal value" r_0 if confidence is 0 and equal to a "maximal value" r_1 if confidence is 1. Matrices r_0 and r_1 are chosen to be constant and invertible. For completeness, f is modeled as:

$$f(\mathbf{x}, t) = f_{\text{sensor}}(\mathbf{x}, t) f_{\text{noise}}(\mathbf{x}, t) f_{\mathbb{H}}(\mathbf{x}, t) \tag{35}$$

f_{sensor} indicates the availability of observation values: it is equal to 0 if data are not or wrongly acquired by the sensor. f_{noise} characterizes data quality: it is close to 0 for noisy data. $f_{\mathbb{H}}$ measures the confidence in the observation model; it is close to 0 if the observation equation is not valid.

With such definition of R, observation values with a low confidence will not be considered during the computation of the solution of the Image Processing problem.

4.4 Conclusion

In this Section, we considered the class of ill-posed Image Processing problems which are usually solved by constraining spatially and/or temporally the solution. We proposed a generic method to convert the Image Model and the spatio-temporal regularity constraint on the result in term of Data Assimilation components, as expressed in equations (5,6,8). The Image Model is taken as observation model and the observation error weights the importance of the observation values in the computation of the state vector: when data are irrelevant, their values are no more used. Of course, this is only possible if the quality of the observations can be evaluated. The evolution equation describes the temporal evolution of the state vector and we proposed two generic choices for the operator $\mathbb{M}$. However, both are too restrictive to deal with complex temporal dynamics and prior knowledge should be included when available. The coherency of the temporal evolution through the operator $\mathbb{M}$ is ensured by the covariance matrix Q and we gave three examples of regularization. In the next Section, we illustrate how to use Data Assimilation to determine optical flow by applying these general principles.

5 Application to optical flow estimation

Let I be a sequence of images on a bounded domain of $\mathbb{R}^2$, denoted Ω. Let $\mathbf{W}(\mathbf{x},t)$ be the velocity vector of a point $\mathbf{x}\in\Omega$ between t and $t+\triangle t$, verifying:

$$I(\mathbf{x}+\mathbf{W}(\mathbf{x},t)\triangle t, t+\triangle t) = I(\mathbf{x},t) \tag{36}$$

As this equation is non linear with respect to $\mathbf{W}$, the left member of equation (36) is often linearized using a first order Taylor development around $\triangle t = 0$. This provides the so-called optical flow constraint equation [6]:

$$\nabla I^T(\mathbf{x},t)\mathbf{W}(\mathbf{x},t) + \frac{\partial I}{\partial t}(\mathbf{x},t) = 0 \quad \forall \mathbf{x}\in\Omega \tag{37}$$

By identifying (37) with (1), we have $\mathbf{X}=\mathbf{W}$, $\mathbf{Y}=(\nabla I, I_t)$ and $\mathbb{I} = \nabla I^T(\mathbf{x},t)\mathbf{W}(\mathbf{x},t) + \frac{\partial I}{\partial t}(\mathbf{x},t)$. Estimating apparent motion is an ill-posed problem: the velocity vector has two components and the optical flow equation is not sufficient to compute both. A solution could be obtained in the Image Processing context using a Tikhonov regularization as stated in Section 2, by constraining spatially [6] or spatially and temporally [26] the solution. Equation (37) is often preferred to (36) because it is linear and the associated Euler-Lagrange equation can be discretized using an explicit and robust numerical scheme. However, it is possible to directly use the equation (36) for estimating the optical flow. This has been described in [3] where the resulting Euler-Lagrange equation has been approximated with a semi-implicit scheme. The advantage of equation (36) is that $\mathbf{W}$ can be estimated even for high velocity norms. Equation (37) is an approximation and only well suited for low velocity except if incremental algorithms [12,19] or scale-space methods [1] are considered.

In this paper, we choose however to consider the optical flow constraint (37) in order to illustrate the tasks to be applied for going from an ill-posed Image Processing problem to a Data Assimilation system, compare it with state-of-the-art methods, and prove the advantage of Data Assimilation when processing noisy acquisitions including missing data.

5.1 Observation and evolution models

The optical flow constraint is chosen as image model. As $\mathbb{H}\equiv\mathbb{I}$, the observation model is:

$$\mathbb{H}(\mathbf{W},I)(\mathbf{x},t) = \nabla I(\mathbf{x},t)^T\mathbf{W}(\mathbf{x},t) + I_t(\mathbf{x},t) \tag{38}$$

With this choice, the field $\mathbf{W}(\mathbf{x},t)$ of velocity vectors is now considered as the state vector $\mathbf{X}(\mathbf{x},t)$ and the image gradients $(\nabla I(\mathbf{x},t), I_t(\mathbf{x},t))$ constitutes the observation vector $\mathbf{Y}(\mathbf{x},t)$.

We have to define an appropriate observation error. Equations (33) and (34) are used to define the inverse of R and to locate the observation values which must be discarded. The observation model $\mathbb{H}$ being scalar, the matrices r_0 and r_1 are scalars and respectively set to ϵ and $1-\epsilon$ with $\epsilon \simeq 10^{-6}$. Equation (35) is used as the observation confidence. Without any information, f_{noise} is assumed to be equal to 1. f_{sensor} is set to 0 if data are not acquired and to 1 otherwise. $f_{\mathbb{H}}$ is chosen from the following remark: the spatio-temporal gradient is null on regions of uniform grey level

values and equation (37) is then degenerated on pixels belonging to them. For avoiding further considering these points, $f_{\mathbb{H}}$ is defined by:

$$f_{\mathbb{H}}(\mathbf{x},t) = 1 - \exp(-\|\nabla_3 I(\mathbf{x},t)\|^2) \tag{39}$$

where ∇_3 denotes the spatio-temporal gradient operator.

The transport of the velocity, equation (32), is taken as evolution equation:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \mathbf{W}^T \mathbf{W} = 0 \tag{40}$$

This equation is rewritten as a two-component system:

$$\frac{\partial U}{\partial t} + UU_x + VU_y = 0 \tag{41}$$

$$\frac{\partial V}{\partial t} + UV_x + VV_y = 0 \tag{42}$$

and by identification of equation (5) with (40) and (41,42), the evolution model is:

$$\begin{aligned}\mathbb{M}(\mathbf{W}) &= \begin{pmatrix}\mathbb{M}_1(\mathbf{W}) & \mathbb{M}_2(\mathbf{W})\end{pmatrix}^T \\ &= \begin{pmatrix}UU_x + VU_y & UV_x + VV_y\end{pmatrix}^T\end{aligned}$$

$\mathbf{W}$ being a two-component vector, the Q matrix is of size 2×2 and chosen as:

$$Q(\mathbf{x},t,\mathbf{x}',t') = q(\mathbf{x}-\mathbf{x}',t-t') \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} \tag{43}$$

with $q(\mathbf{x},t) = \exp(-\frac{1}{\sigma}(\|\mathbf{x}\|+|t|))$ or $q(\mathbf{x},t) = \delta(\mathbf{x})\delta(t)$. We have tested both covariance matrices with $\sigma = 1$ without noticing significant differences on the results.

The background value at $t = 0$ (equation (6)) has also to be provided: we make use of Horn and Schunck's algorithm [6] to compute the velocity field on the two first frames of the sequence. We consider the background error $B(\mathbf{x},\mathbf{x}') = \delta(\mathbf{x}-\mathbf{x}')$ for measuring the quadratic discrepancy between $\mathbf{X}_b(\mathbf{x})$ and $\mathbf{X}(\mathbf{x},0)$.

5.2 Adjoint operators

In order to determine the adjoint operators for $\mathbb{M}$ and $\mathbb{H}$, the directional derivatives must first be established.
Using the definition (11), we obtain:

$$\begin{aligned}\frac{\partial \mathbb{M}_1}{\partial \mathbf{W}}(\eta) &= \frac{\partial \mathbb{M}_1}{\partial U}(\eta^1) + \frac{\partial \mathbb{M}_1}{\partial V}(\eta^2) = U\eta^1_x + U_x\eta^1 + V\eta^1_y + U_y\eta^2 \\ \frac{\partial \mathbb{M}_2}{\partial \mathbf{W}}(\eta) &= \frac{\partial \mathbb{M}_2}{\partial U}(\eta^1) + \frac{\partial \mathbb{M}_2}{\partial V}(\eta^2) = U\eta^2_x + V_y\eta^2 + V\eta^2_y + V_x\eta^1\end{aligned}$$

with $\eta = \begin{pmatrix}\eta^1 & \eta^2\end{pmatrix}^T$ and η_x and η_y standing for partial derivate with respect to x and y. The reader is referred to Appendix B.1 for more details. Using definition (17),

integration by parts and considering boundary terms equal to zero, the adjoint operator of $\mathbb{M}$ is:

$$\left(\frac{\partial \mathbb{M}_1}{\partial \mathbf{W}}\right)^* (\lambda) = -U\lambda_x^1 - V_y\lambda^1 - V\lambda_y^1 + V_x\lambda^2$$
$$\left(\frac{\partial \mathbb{M}_2}{\partial \mathbf{W}}\right)^* (\lambda) = -U_x\lambda^2 - U\lambda_x^2 - V\lambda_y^2 + U_y\lambda^1$$

with $\lambda = \begin{pmatrix}\lambda^1 & \lambda^2\end{pmatrix}^T$. Details are reported in Appendix B.2. In a compact form, the adjoint operator of $\mathbb{M}$ is written:

$$\begin{aligned}\left(\frac{\partial \mathbb{M}}{\partial \mathbf{W}}\right)^* (\lambda) &= -\begin{pmatrix}\lambda_x^1 & \lambda_x^2\\ \lambda_y^1 & \lambda_y^2\end{pmatrix}^T \begin{pmatrix}U\\ V\end{pmatrix} - \begin{pmatrix}V_y & -U_y\\ -V_x & U_x\end{pmatrix}^T \begin{pmatrix}\lambda^1\\ \lambda^2\end{pmatrix}\\ &= -\begin{pmatrix}\nabla\lambda_1 & \nabla\lambda_2\end{pmatrix}^T \mathbf{W} - \begin{pmatrix}\nabla^\perp V & -\nabla^\perp U\end{pmatrix}^T \lambda\end{aligned}$$

with $\nabla^\perp U = \begin{pmatrix}U_y & -U_x\end{pmatrix}^T$.
The directional derivative of the observation operator is:

$$\frac{\partial \mathbb{H}}{\partial \mathbf{W}}(\eta)(\mathbf{x},t) = \nabla I^T(\mathbf{x},t)\eta(\mathbf{x},t)$$

and determining the adjoint operator is direct, as described in Appendix B.3:

$$\left(\frac{\partial \mathbb{H}}{\partial \mathbf{W}}\right)^* (\lambda)(\mathbf{x},t) = \nabla I(\mathbf{x},t)\lambda(\mathbf{x},t)$$

5.3 Discretization

Using the choices made in Subsection 5.1, differential and adjoint operators values calculated in Subsection 5.2, the three PDEs (23,21,25) become:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla\mathbf{W}^T\mathbf{W} = 0 \tag{44}$$

$$-\frac{\partial \lambda}{\partial t} - \nabla\lambda^T\mathbf{W} - (\nabla^\perp\mathbf{W})^T\lambda = -\nabla I R^{-1} \star L \tag{45}$$

$$\frac{\partial \delta\mathbf{W}}{\partial t} + \nabla\delta\mathbf{W}^T\mathbf{W} + \nabla\mathbf{W}^T\delta\mathbf{W} = Q \star \lambda \tag{46}$$

with $\nabla^\perp\mathbf{W} = \begin{pmatrix}\nabla^\perp V & -\nabla^\perp U\end{pmatrix}$, and $L = I_t + \nabla I^T(\mathbf{W} + \delta\mathbf{W})$. The covariance matrices Q and R being chosen isotropic, they only depend on $\mathbf{x} - \mathbf{x}'$ and $t - t'$, and the right members of equations (45) and (46) can then be expressed as a convolution product. For evaluating these right members at a given date, the knowledge of L and λ over the whole temporal domain is required. However, choosing a Dirac matrix for Q and R^{-1} reduces the convolution product to a simple multiplication. In (45) computation of $\lambda(t)$ is only depending on values at $(t+1)$ and in (46) $\delta\mathbf{X}(t)$ is obtained from variables at $(t-1)$. The algorithm becomes a frame-by-frame process: the whole sequence of state and observation vectors, adjoint and incremental variables may be left on the mass storage, excepted frames implied in the computation.

The three equations are discretized using a finite difference technique. Let us first examine equation (44): it is a 2D non linear advection equation. The advection term

corresponds to the velocity transport. Its direct approximation by a standard Euler scheme is known as being numerically unstable. This instability has several origins: the non-linearity of the equation, the multi-dimensionality of the state vector, the simultaneous occurrence of linear and non linear terms in the equation. For stabilizing the scheme, a diffusive term is often introduced into the equation:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \mathbf{W}^T \mathbf{W} = \kappa \nabla^2 \mathbf{W}$$

with $\kappa > 0$ having a small value as done in [15]. This is known under the name of Lax-Wendroff method. Such an equation can then be approximated using an explicit Euler scheme (with Courant-Friedrich-Levy condition) or an implicit Euler scheme. This has the drawback of smoothing the solution and not preserving sharp discontinuities. We propose, in the following, a stable scheme for the advection equation, without adding this diffusive term, by using a splitting method [23]. As $\mathbf{W}(\mathbf{x}, t)$ is a vector of $\mathbb{R}^2$, equation (44) has two components (41) and (42). The first one combines a term of linear advection in direction y and non linear one in direction x and is expressed as a two-equation system using the splitting method:

$$\frac{\partial U}{\partial t} + UU_x = 0 \tag{47}$$

$$\frac{\partial U}{\partial t} + VU_y = 0 \tag{48}$$

Equation (47) is rewritten with the Lax-Friedrich method [20] as $\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0$ with $F(U) = \frac{1}{2}U^2$. This new equation is discretized by:

$$U_{i,j}^{k+1} = \frac{1}{2}(U_{i+1,j}^k + U_{i-1,j}^k) - \frac{\triangle t}{2}(F_{i+1,j}^k - F_{i-1,j}^k)$$

with $U_{i,j}^k = U(x_i, y_i, t_k)$, $F_{i,j}^k = F(U(x_i, y_i, t_k))$ and $\triangle t$ the time step. The term $\frac{1}{2}(U_{i+1,j}^k + U_{i-1,j}^k)$ stabilizes the scheme by adding a diffusive effect while $\triangle t$ satisfies the Courant-Friedrich-Levy condition. The linear advection (48) is discretized using an explicit shock scheme [20]:

$$U_{i,j}^{k+1} = U_{i,j}^k - \triangle t \left(\max(V_{i,j}^k, 0) \left(U_{i,j}^k - U_{i,j-1}^k \right) + \min(V_{i,j}^k, 0) \left(U_{i,j+1}^k - U_{i,j}^k \right) \right)$$

In the same way, it can be seen that the second component of (44) contains a linear advection term in direction x and a non linear one in direction y. The same strategy is then applied for discretization.

Equation (45) combines a linear advection ($\nabla \lambda^T \mathbf{W}$), a term of reaction ($(\nabla^\perp \mathbf{W})^T \lambda$) and a forcing term ($\nabla I R^{-1} \star L$). Its first component is $-\frac{\partial \lambda^1}{\partial t} - U\lambda_x^1 - V_y\lambda^1 - V\lambda_y^1 + V_x\lambda^2 = \frac{\partial I}{\partial x} A$ with $A = -R^{-1} \star L$. It is split into two parts. The first part contains the linear advection in direction x and the reaction term: $-\frac{\partial \lambda^1}{\partial t} - U\lambda_x^1 - V_y\lambda^1 = 0$ and is discretized in the same way as (48) with an explicit shock scheme. However, the equation is retrograde and its initial condition is given at time T:

$$(\lambda^1)_{i,j}^{k-1} = \left(1 + \frac{\triangle t}{2}(V_{i,j+1}^k - V_{i,j-1}^k)\right)(\lambda^1)_{i,j}^k + \triangle t \left(\max(U_{i,j}^k, 0)((\lambda^1)_{i,j}^k - (\lambda^1)_{i-1,j}^k) + \min(U_{i,j}^k, 0)((\lambda^1)_{i+1,j}^k - (\lambda^1)_{i,j}^k) \right)$$

The second part contains the linear advection term in direction y and the forcing term: $-\frac{\partial \lambda^1}{\partial t} - V\lambda^1_y = -V_x\lambda^2 + \frac{\partial I}{\partial x}A$. Again, an explicit shock scheme is used:

$$(\lambda^1)^{k-1}_{i,j} = (\lambda^1)^k_{i,j} - \frac{\triangle t}{2}\left(V^k_{i+1,j} - V^k_{i-1,j}\right)(\lambda^2)^k_{i,j} + \triangle t(I_x A)^k_{i,j} + \\ \triangle t\left(\max(V^k_{i,j},0)((\lambda^1)^k_{i,j} - (\lambda^1)^k_{i,j-1}) + \min(V^k_{i,j},0)((\lambda^1)^k_{i,j+1} - (\lambda^1)^k_{i,j})\right)$$

Having the same structure, the second component of (45) is discretized with the same method. The complete numerical scheme is described in Appendix C.2.

The last equation, (46), is similar to equation (45): a linear advection with a reaction term and a forcing term. We therefore use the same discretization technique. The full numerical scheme is detailed in Appendix C.3.

5.4 Results

The "taxi" sequence and a synthetic sequence have been chosen for discussing results. In both cases, image gradients are computed with a convolution method and a derivative Gaussian kernel whose variance is set to 1. The incremental algorithm (Subsection 3.3) is iterated 5 times.

The taxi sequence displays several cars moving with a slow and quasi uniform motion.
In a first experiment, we compute the optical flow using the Data Assimilation method with image gradients as observations. Horn & Schunk's method is also applied on the sequence and both results are displayed for comparison and analysis purposes of our method. Figures 1 to 3 show the results obtained on three frames of the sequence. These results are qualitatively similar, illustrating that both methods are equivalent when observation values are available on the whole sequence. Having chosen Horn & Schunck or another image processing method does not really matter, because we are not interested in discussing a quantitative comparison of optical flow methods but in proving the efficiency of Data Assimilation for dealing with missing data and complex dynamics. Consequently, a second experiment is designed for analyzing the issue of missing data. A large region around the white car, denoted $\Re$, is set to zero (black rectangle) on one frame of the sequence to simulate a sensor failure. To indicate the irrelevance of pixel values inside this region, the function f_{sensor} returns the value 0 inside $\Re$ and the value 1 outside. The spatio-temporal gradient is then computed on the modified sequence and provides the observations. Figure 4 shows the results with Data Assimilation and Horn & Schunk methods. This latter obviously fails to provide acceptable velocity vectors over $\Re$, while Data Assimilation provides a correct result thanks to the eviction of missing observation in the computation and to the evolution equation. A similar experiment is performed by setting several small regions to zero on one frame and flagged them as not acquired with f_{sensor} equal to zero. This lack of observation also disturbs Horn & Schunk's algorithm while Data Assimilation provides a correct result as illustrated by Figure 5. Even a whole frame of the observation sequence can be missing: we force image gradients to zero on the fifth frame of the taxi sequence resulting to $f_{\mathbb{H}} = 0$ on this frame. In this case, Horn & Schunk's method can not provide any result on this frame. Figure 6 is then comparing results obtained by Data Assimilation with and without observation on frame 5. Results remain similar,

due to the fact that the evolution model correctly approximates the temporal dynamics of these data and compensates the missing acquisition.

Fig. 1 Comparison Data Assimilation (left) / Horn-Schunk (right) – frame 3.

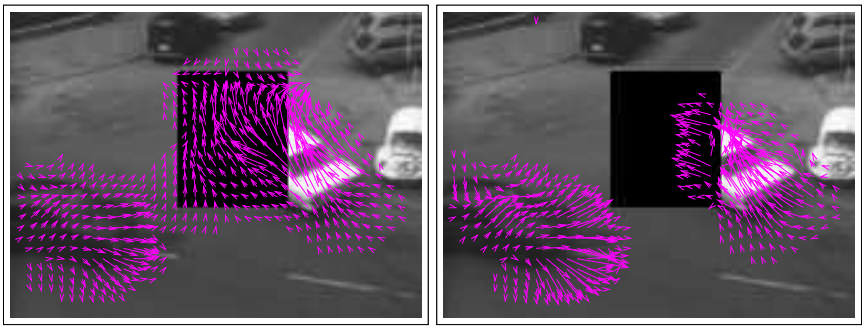
Fig. 2 Comparison Data Assimilation (left) / Horn-Schunk (right) – frame 6.

A third experiment is dedicated to prove that using the real dynamics within the evolution model ensures getting correct estimation of motion. For that purpose, we built a synthetic sequence displaying one square moving horizontally from the left up part of the image and one moving vertically from the right bottom part. At the end of the sequence, the two squares meet each other. Figures 7(b), 8(b) and 9(b) show the results with Horn & Schunk's method which fails to estimate a correct velocity direction when the squares meet. This is due to an over-regularization by the cost function. Figures 7(a), 8(a) and 9(a) show the results with Data Assimilation: motion is better estimated because the evolution model correctly describes the dynamics and avoids the negative effects of spatial regularization.

6 Conclusion

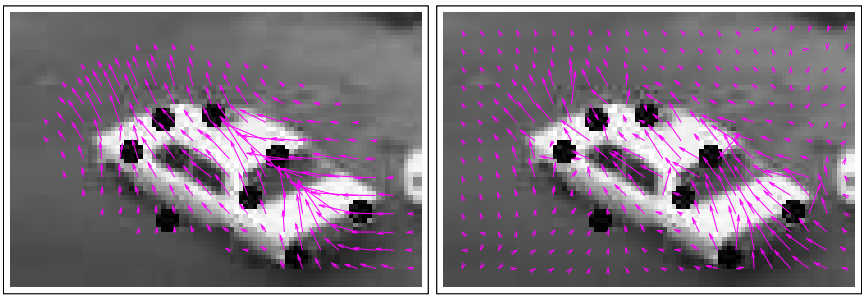
In this paper we proposed a general framework to solve ill-posed Image Processing problems by Data Assimilation methods. This is an alternative to the space-time ap-

Fig. 3 Comparison Data Assimilation (left) / Horn-Schunk (right) – frame 9.



(a) Data assimilation (b) Horn-Schunk

Fig. 4 Missing data on a large region in frame 5.

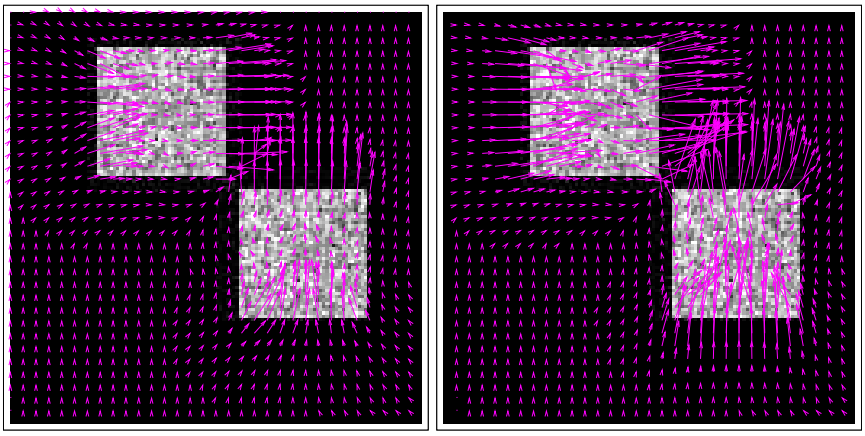


(a) Data assimilation (b) Horn-Schunk

Fig. 5 Missing data on small regions in frame 5.

(a) Result with image gradients set to 0 on frame 5 (b) Result with image gradient available

Fig. 6 Missing data on the whole frame 5.



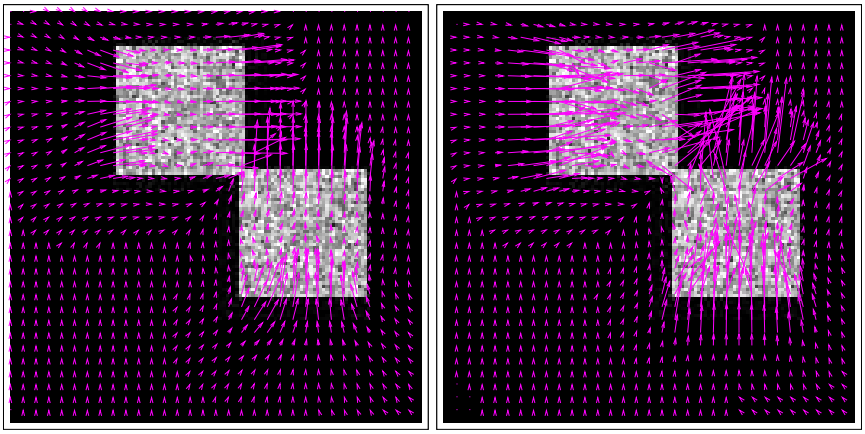
(a) Data Assimilation (b) Horn-Schunk

Fig. 7 Results on synthetic sequence - frame 4.

proach, described in the introduction, which constrains the solution's variations in space and time.

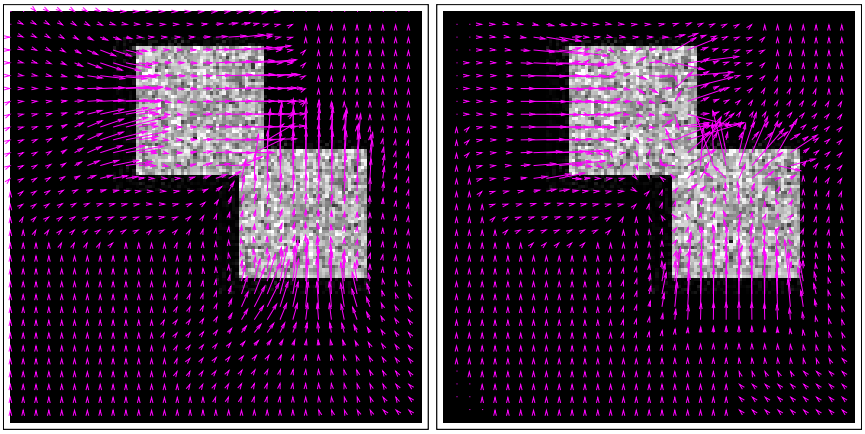
If the dynamics is approximately known and expressed as an evolution equation, we show how this information is used, simultaneously with the observation equation, in the framework of Data Assimilation, to temporally constrain the solution and obtain a better result. We extensively describe how to formalize the image processing problem using the Data Assimilation framework.

The impact of covariance matrices on the energy minimization has been investigated. Their regularization properties have been described for three different cases. The knowledge of the temporal dynamics and the choice of relevant covariance matrices make it possible to handle the problem of missing and noisy data. For that purpose, the observation error is described by a specific covariance matrix, with high values on pixels corresponding to missing and noisy data, which are then discarded during the com-



(a) Data Assimilation (b) Horn-Schunk

Fig. 8 Results on synthetic sequence - frame 7.



(a) Data Assimilation (b) Horn-Schunk

Fig. 9 Results on synthetic sequence - frame 10.

putation process. On these pixels, the solution is mainly obtained by the evolution equation. Obviously, this is only possible if a confidence measure on observation data is available.

Due to specific choices of covariances, the 4D-Var algorithm described in the paper allows a local computation of the state vector, on a frame-by-frame basis. It therefore has memory management advantages compared to the space-time approach, which requires to include the whole sequence in memory, as the linear system is solved in the space-time domain.

The general approach has been illustrated on estimation of the optical flow by assimilating image gradient observations in a model, that describes the evolution of

velocity by itself. This transport equation is simple and correctly approximates the dynamics, which is locally translational and uniform. The chosen observation equation is the optical flow constraint, linearly linking velocity and image brightness gradients. The discretization leads to an unstable numerical scheme if standard finite difference methods are used. To overcome this problem, a robust scheme, based on a splitting method, has been proposed in the paper.

A first perspective of this work is to investigate the evolution equation issue. For some applications it is possible to use a physically consistent equation: for instance, ocean surface motion is described with a shallow-water model. If we are concerned with video sequences of rigid objects, the transport of velocity by itself is relevant for locally translational displacements, but becomes inadequate for rotational or discontinuous motion. A solution is to build a parametric model of the image dynamics: the spatio-temporal domain is first divided in sub-domains and the transport equation is assumed to correctly describe the dynamics on each of them.
A second perspective is to investigate the spatial regularization of the state vector $\mathbf{X}$ through the observation covariance matrix R.
A last perspective addresses the experimental application chosen in the paper: the estimation of optical flow. The optical flow equation has been used as observation equation. This equation has the advantage to be linear but is only an approximation of the transport of image brightness by velocity. The initial transport equation, even if non linear, can be used as observation equation. The differential and adjoint operator of the observation model can be determined if the image brightness is supposed to be differentiable. The 4D-Var algorithm, described in Subsection 3.3, could then be used with such observation equation.

A The Euler-Lagrange equation of E (10)

Let us first compute the derivative of E in direction η using definition (11):

$$\begin{aligned}
E(\mathbf{X}+\gamma\eta) = & \frac{1}{2}\int_A\int_A (\mathbf{X}_t+\gamma\eta_t+\mathbb{M}(\mathbf{X}+\gamma\eta))^T Q^{-1} (\mathbf{X}_t+\gamma\eta_t+\mathbb{M}(\mathbf{X}+\gamma\eta))^T d\mathbf{x}dtd\mathbf{x}'dt' \\
& + \int_A\int_A \mathbb{H}(\mathbf{X}+\gamma\eta,\mathbf{Y})^T R^{-1}\mathbb{H}(\mathbf{X}+\gamma\eta,\mathbf{Y})d\mathbf{x}dtd\mathbf{x}'dt' \\
& + \int_\Omega\int_\Omega (\mathbf{X}+\gamma\eta-\mathbf{X}_b)^T B^{-1}(\mathbf{X}+\gamma\eta-\mathbf{X}_b)d\mathbf{x}d\mathbf{x}' \\
\frac{d}{d\gamma}E(\mathbf{X}+\gamma\eta) = & \int_A\int_A \left(\eta_t+\frac{d}{d\gamma}\mathbb{M}(\mathbf{X}+\gamma\eta)\right)^T Q^{-1} (\mathbf{X}_t+\gamma\eta_t+\mathbb{M}(\mathbf{X}+\gamma\eta)) d\mathbf{x}dtd\mathbf{x}'dt' \\
& + \int_A\int_A \frac{d}{d\gamma}\left(\mathbb{H}(\mathbf{X}+\gamma\eta,\mathbf{Y})^T\right) R^{-1}\mathbb{H}(\mathbf{X}+\gamma\eta,\mathbf{Y})d\mathbf{x}dtd\mathbf{x}'dt' \\
& + \int_\Omega\int_\Omega \eta^T B^{-1}(\mathbf{X}+\gamma\eta-\mathbf{X}_b)d\mathbf{x}d\mathbf{x}'
\end{aligned}$$

Let us γ tend to zero:

$$\begin{aligned}
\frac{\partial E}{\partial \mathbf{X}}(\eta) = & \int_A\int_A \left(\eta_t+\frac{\partial \mathbb{M}}{\partial \mathbf{X}}(\eta)\right)^T Q^{-1} (\mathbf{X}_t+\mathbb{M}(\mathbf{X})) d\mathbf{x}dtd\mathbf{x}'dt' \\
& + \int_A\int_A \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}(\eta)\right)^T R^{-1}\mathbb{H}(\mathbf{X},\mathbf{Y})d\mathbf{x}dtd\mathbf{x}'dt' \\
& + \int_\Omega\int_\Omega \eta^T B^{-1}(\mathbf{X}-\mathbf{X}_b)d\mathbf{x}d\mathbf{x}'
\end{aligned}$$

We use integration by parts in order to factorize each term with η^T:

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{X}}(\eta) = &\int_A\int_A \eta^T \left(\delta(t=\mathbf{T}) - \delta(t=0) - \frac{\partial}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^*\right) Q^{-1}\left(\mathbf{X}_t + \mathbb{M}(\mathbf{X})\right) d\mathbf{x}dtd\mathbf{x}'dt' \\ &+ \int_A\int_A \eta^T \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* R^{-1}\mathbb{H}(\mathbf{X},\mathbf{Y}) d\mathbf{x}dtd\mathbf{x}'dt' \\ &+ \int_\Omega\int_\Omega \eta^T B^{-1}(\mathbf{X}-\mathbf{X}_b) d\mathbf{x}d\mathbf{x}'\end{aligned}$$

Let us introduce the adjoint variable λ in the previous expression and use Fubbini's theorem:

$$\begin{aligned}\frac{\partial E}{\partial \mathbf{X}}(\eta) = &\int_\Omega \eta^T(\mathbf{x},\mathbf{T})\lambda(\mathbf{x},\mathbf{T})d\mathbf{x} - \int_\Omega \eta^T(\mathbf{x},0)\lambda(\mathbf{x},0)d\mathbf{x} \\ &+ \int_A \eta^T(\mathbf{x},t)\left(-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^*(\lambda)\right) d\mathbf{x}dt \\ &+ \int_A \eta^T(\mathbf{x},t)\left(\int_A \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* R^{-1}\mathbb{H}(\mathbf{X},\mathbf{Y})d\mathbf{x}'dt'\right) d\mathbf{x}dt \\ &+ \int_\Omega \eta^T(\mathbf{x},0)\left(\int_\Omega B^{-1}(\mathbf{X}(\mathbf{x}',0)-\mathbf{X}_b(\mathbf{x}'))d\mathbf{x}'\right) d\mathbf{x}\end{aligned}$$

A solution of $\dfrac{\partial E}{\partial \mathbf{X}}(\eta) = 0 \,\forall \eta$ w.r.t. $\mathbf{X}$ verifies the following system:

$$\lambda(\mathbf{x},\mathbf{T}) = 0 \tag{49}$$

$$-\lambda(\mathbf{x},0) + \int_\Omega B^{-1}(\mathbf{X}(\mathbf{x}',0)-\mathbf{X}_b(\mathbf{x}'))d\mathbf{x}' = 0 \tag{50}$$

$$-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\right)^*(\lambda) + \int_A \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\right)^* R^{-1}\mathbb{H}(\mathbf{X},\mathbf{Y})d\mathbf{x}'dt' = 0 \tag{51}$$

Using the definition of inverse covariance (26), Equation (50) is rewritten as:

$$\mathbf{X}(\mathbf{x},0) = \mathbf{X}_b(\mathbf{x}) + \int_\Omega B(\mathbf{x},\mathbf{x}')\lambda(\mathbf{x}',0)d\mathbf{x}' \tag{52}$$

and the state vector is expressed from λ using (12):

$$\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}) = \int_A Q(\mathbf{x},t,\mathbf{x}',t')\lambda(\mathbf{x}',t')d\mathbf{x}'dt' \tag{53}$$

Equations (49), (51), (52) and (53) are the Euler-Lagrange equations.

B Determination of the adjoint operators

B.1 Differential of $\mathbb{M}$

The operator $\mathbb{M}$ is defined by:

$$\mathbb{M}(\mathbf{W}) = \mathbf{W}^T\nabla\mathbf{W} = \begin{pmatrix}\mathbb{M}_1(\mathbf{W})\\ \mathbb{M}_2(\mathbf{W})\end{pmatrix} = \begin{pmatrix}UU_x + VU_y\\ UV_x + VV_y\end{pmatrix}$$

The differential of $\mathbb{M}$ is formally equal to the following Jacobian matrix:

$$\frac{\partial \mathbb{M}}{\partial \mathbf{W}} = \begin{pmatrix}\dfrac{\partial \mathbb{M}_1}{\partial U} & \dfrac{\partial \mathbb{M}_1}{\partial V}\\ \dfrac{\partial \mathbb{M}_2}{\partial U} & \dfrac{\partial \mathbb{M}_2}{\partial V}\end{pmatrix} \tag{54}$$

Each element of this matrix is obtained using the definition of the directional derivative. Considering the first element of $\frac{\partial \mathbb{M}}{\partial \mathbf{W}}$, the directional derivative with respect to U in direction η^1 is given by:

$$\left(\frac{\partial \mathbb{M}_1}{\partial U}\right)^T \eta^1 = \lim_{\gamma \to 0} \frac{d}{d\gamma} \left(\mathbb{M}_1(U + \gamma\eta^1, V)\right)$$

and we obtain:

$$\begin{aligned}
\frac{\partial \mathbb{M}_1}{\partial U}^T \eta^1 &= U\eta^1_x + V\eta^1_y + U_x\eta^1 \\
\frac{\partial \mathbb{M}_1}{\partial V}^T \eta^2 &= U_y\eta^2 \\
\frac{\partial \mathbb{M}_2}{\partial U}^T \eta^1 &= V_x\eta^1 \\
\frac{\partial \mathbb{M}_2}{\partial V}^T \eta^2 &= U\eta^2_x + V\eta^2_y + V_y\eta^2
\end{aligned}$$

$$\frac{\partial \mathbb{M}}{\partial \mathbf{W}} = \begin{pmatrix} U\partial_x + V\partial_y + U_x & U_y \\ V_x & U\partial_x + V\partial_y + V_y \end{pmatrix}$$

In equation (25), the differential of $\mathbb{M}$ appears as a directional derivative in direction $\delta\mathbf{W}$ and it has been defined in such a way (see equation (54)) that the Jacobian matrix does not need to be transposed to compute the derivative in a given direction. It is therefore evaluated as:

$$\begin{aligned}
\frac{\partial \mathbb{M}}{\partial \mathbf{W}}(\delta\mathbf{W}) = \frac{\partial \mathbb{M}}{\partial \mathbf{W}}\delta\mathbf{W} &= \begin{pmatrix} \frac{\partial \mathbb{M}_1}{\partial U}\delta U + \frac{\partial \mathbb{M}_1}{\partial V}\delta V \\ \frac{\partial \mathbb{M}_2}{\partial U}\delta U + \frac{\partial \mathbb{M}_2}{\partial V}\delta V \end{pmatrix} \\
&= \begin{pmatrix} U\delta U_x + V\delta U_y + U_x\delta U + U_y\delta V \\ U\delta V_x + V\delta V_y + V_x\delta U + V_y\delta V \end{pmatrix}
\end{aligned}$$

B.2 Adjoint operator of $\frac{\partial \mathbb{M}}{\partial \mathbf{W}}$

The adjoint operator of $\frac{\partial \mathbb{M}}{\partial \mathbf{W}}$ is formally defined by:

$$\begin{aligned}
\left\langle \frac{\partial \mathbb{M}}{\partial \mathbf{W}}(\delta\mathbf{W}), \lambda \right\rangle_{L^2} &= \left\langle \delta\mathbf{W}, \left(\frac{\partial \mathbb{M}}{\partial \mathbf{W}}\right)^* (\lambda) \right\rangle_{L^2} \\
&= \int \left(\frac{\partial \mathbb{M}}{\partial \mathbf{W}}(\delta\mathbf{W})\right)^T \lambda d\mathbf{x}
\end{aligned}$$

The directional derivative $\frac{\partial \mathbb{M}}{\partial \mathbf{W}}(\delta\mathbf{W})$ is a column vector and has to be transposed to perform the scalar product with $\lambda = \begin{pmatrix}\lambda^1 & \lambda^2\end{pmatrix}^T$:

$$\begin{aligned}
\int \left(\frac{\partial \mathbb{M}}{\partial \mathbf{W}}(\delta\mathbf{W})\right)^T \lambda d\mathbf{x} &= \int \begin{pmatrix} U\delta U_x + V\delta U_y + U_x\delta U + U_y\delta V \\ U\delta V_x + V\delta V_y + V_x\delta U + V_y\delta V \end{pmatrix}^T \begin{pmatrix}\lambda^1 \\ \lambda^2\end{pmatrix} d\mathbf{x} \\
&= \int \Big(-\delta U\partial_x(U\lambda^1) - \delta U\partial_y(V\lambda^1) + \delta U U_x\lambda^1 + \delta V U_y\lambda^1 \\
&\qquad \delta U V_x\lambda^2 - \delta V\partial_x(U\lambda^2) - \delta V\partial_y(V\lambda^2) + \delta V V_y\lambda^2 \Big) d\mathbf{x} \\
&= \int \begin{pmatrix}\delta U & \delta V\end{pmatrix} \begin{pmatrix} -U\lambda^1_x - V_y\lambda^1 - V\lambda^1_y + V_x\lambda^2 \\ -U_x\lambda^2 - U\lambda^2_x - V\lambda^2_y + U_y\lambda^1 \end{pmatrix} d\mathbf{x} \\
&= \int \begin{pmatrix}\delta U & \delta V\end{pmatrix} \begin{pmatrix} -U\partial_x - V\partial_y - V_y & V_x \\ U_y & -U\partial_x - V\partial_y - U_x \end{pmatrix} \begin{pmatrix}\lambda^1 \\ \lambda^2\end{pmatrix} d\mathbf{x}
\end{aligned}$$

The adjoint operator of $\dfrac{\partial \mathbb{M}}{\partial \mathbf{W}}$ is:

$$\left(\frac{\partial \mathbb{M}}{\partial \mathbf{W}}\right)^* = \begin{pmatrix} -U\partial_x - V\partial_y - V_y & V_x \\ U_y & -U\partial_x - V\partial_y - U_x \end{pmatrix}$$

and its expression in direction λ is:

$$\left(\frac{\partial \mathbb{M}}{\partial \mathbf{W}}\right)^* (\lambda) = \begin{pmatrix} -U\lambda_x^1 - V\lambda_y^1 - V_y\lambda^1 + V_x\lambda^2 \\ U_y\lambda^1 - U\lambda_x^2 - V\lambda_y^2 - U_x\lambda^2 \end{pmatrix}$$

B.3 Differential of $\mathbb{H}$ and adjoint operator

The operator $\mathbb{H}$ is defined by:

$$\mathbb{H}(\mathbf{W}, I)(\mathbf{x}, t) = \nabla I(\mathbf{x}, t)^T \mathbf{W}(\mathbf{x}, t) + I_t(\mathbf{x}, t)$$

The differential of $\mathbb{H}$ is:

$$\frac{\partial \mathbb{H}}{\partial \mathbf{W}} = \begin{pmatrix} \dfrac{\partial \mathbb{H}}{\partial U} & \dfrac{\partial \mathbb{H}}{\partial V} \end{pmatrix}$$

The first component is given by:

$$\begin{aligned} \frac{\partial \mathbb{H}}{\partial U}\eta^1 &= \lim_{\gamma \to 0} \frac{d}{d\gamma} \mathbb{H}(U + \gamma\eta^1, V) \\ &= \lim_{\gamma \to 0} \frac{d}{d\gamma} \left(I_x(U + \gamma\eta^1) + I_y V + I_t\right) \\ &= I_x \eta^1 \end{aligned}$$

The same calculus leads for the second component to $\dfrac{\partial \mathbb{H}}{\partial V}\eta^2 = I_y \eta^2$ and finally the differential of $\mathbb{H}$ is:

$$\frac{\partial \mathbb{H}}{\partial \mathbf{W}} = \nabla I^T$$

The adjoint operator is defined by:

$$\begin{aligned} \left\langle \frac{\partial \mathbb{H}}{\partial \mathbf{W}}(\delta\mathbf{W}), \lambda \right\rangle_{L^2} &= \left\langle \delta\mathbf{W}, \left(\frac{\partial \mathbb{H}}{\partial \mathbf{W}}\right)^* (\lambda) \right\rangle_{L^2} \\ &= \int \nabla I^T \delta\mathbf{W} \lambda d\mathbf{x} \\ &= \int \delta\mathbf{W}^T \nabla I \lambda d\mathbf{x} \end{aligned}$$

And finally, the adjoint operator is:

$$\left(\frac{\partial \mathbb{H}}{\partial \mathbf{W}}\right)^* = \nabla I$$

C Numerical schemes

C.1 Evolution equation of the background variable

Equation (44) has two components and can be written as:

$$\frac{\partial U}{\partial t} + UU_x + VU_y = 0 \tag{55}$$

$$\frac{\partial V}{\partial t} + UV_x + VV_y = 0 \tag{56}$$

Equation (55) is rewritten and split into the system:

$$F = \frac{1}{2}U^2 \tag{57}$$

$$\frac{\partial U}{\partial t} + F_x = 0 \tag{58}$$

$$\frac{\partial U}{\partial t} + VU_y = 0 \tag{59}$$

Equations (58) and (59) are now linear advection equations. They can be approximated using the following schemes:

$$\frac{U_{i,j}^{k+1} - \frac{1}{2}(U_{i+1,j}^k + U_{i-1,j}^k)}{\triangle t} = -\frac{1}{2}(F_{i+1,j}^k - F_{i-1,j}^k)$$

$$\frac{U_{i,j}^{k+1} - U_{i,j}^k}{\triangle t} = -S_y(V,U)_{i,j}^k$$

S is the discrete operator approximating the advection operator using a shock filter, defined in the following in the y direction:

$$S_y(V,U)_{i,j} = \max(V_{i,j},0)\,(U_{i,j} - U_{i,j-1}) + \min(V_{i,j},0)\,(U_{i,j+1} - U_{i,j})$$

Equation (56) is rewritten and split into the system:

$$G = \frac{1}{2}V^2$$

$$\frac{\partial V}{\partial t} + G_y = 0$$

$$\frac{\partial V}{\partial t} + UV_x = 0$$

and then approximated by:

$$\frac{V_{i,j}^{k+1} - \frac{1}{2}(V_{i,j+1}^k + V_{i,j-1}^k)}{\triangle t} = -\frac{1}{2}(G_{i,j+1}^k - G_{i,j-1}^k)$$

$$\frac{V_{i,j}^{k+1} - V_{i,j}^k}{\triangle t} = -S_x(U,V)_{i,j}^k$$

C.2 Evolution equation of the adjoint variable

Equation (45) has two components and can be rewritten as follow:

$$-\lambda_t^1 - U\lambda_x^1 - V_y\lambda^1 - V\lambda_y^1 + V_x\lambda^2 = I_x A \tag{60}$$

$$-\lambda_t^2 - U\lambda_x^2 - U_x\lambda^2 - V\lambda_y^2 + U_y\lambda^1 = I_y A \tag{61}$$

with $A = -R^{-1} \star L$. Equation (60) is split into:

$$-\lambda_t^1 = U\lambda_x^1 + V_y\lambda^1$$

$$-\lambda_t^1 = V\lambda_y^1 - V_x\lambda^2 + I_x A$$

The numerical scheme is retrograde because the initial condition for λ^1 is given at time $t = \mathbf{T}$. We use an explicit Euler scheme:

$$-\frac{(\lambda^1)_{i,j}^k - (\lambda^1)_{i,j}^{k-1}}{\triangle t} = S_x(U,\lambda^1)_{i,j}^k + \frac{1}{2}(V_{i,j+1}^k - V_{i,j-1}^k)(\lambda^1)_{i,j}^k$$

$$-\frac{(\lambda^1)_{i,j}^k - (\lambda^1)_{i,j}^{k-1}}{\triangle t} = S_y(V,\lambda^1)_{i,j}^k - \frac{1}{2}(V_{i+1,j}^k - V_{i-1,j}^k)(\lambda^2)_{i,j}^k + (I_x A)_{i,j}^k$$

The numerical scheme is written:

$$(\lambda^1)^{k-1}_{i,j} = (\lambda^1)^k_{i,j} + \triangle t \left(S_x(U, \lambda^1)^k_{i,j} + \frac{1}{2}(V^k_{i,j+1} - V^k_{i,j-1})(\lambda^1)^k_{i,j} \right)$$
$$(\lambda^1)^{k-1}_{i,j} = (\lambda^1)^k_{i,j} + \triangle t \left(S_y(V, \lambda^1)^k_{i,j} - \frac{1}{2}(V^k_{i+1,j} - V^k_{i-1,j})(\lambda^2)^k_{i,j} + (I_x A)^k_{i,j} \right)$$

The equation (61) is split into:

$$-\lambda^2_t = U\lambda^2_x + U_x\lambda^2$$
$$-\lambda^2_t = V\lambda^2_y - U_y\lambda^1 + (I_y A)$$

and the numerical scheme is:

$$(\lambda^2)^{k-1}_{i,j} = (\lambda^2)^k_{i,j} + \triangle t \left(S_x(U, \lambda^2)^k_{i,j} + \frac{1}{2}(U^k_{i+1,j} - U^k_{i-1,j})(\lambda^2)^k_{i,j} \right)$$
$$(\lambda^2)^{k-1}_{i,j} = (\lambda^2)^k_{i,j} + \triangle t \left(S_y(V, \lambda^2)^k_{i,j} - \frac{1}{2}(U^k_{i,j+1} - U^k_{i,j-1})(\lambda^1)^k_{i,j} + (I_y A)^k_{i,j} \right)$$

C.3 Evolution equation of the incremental variable

Equation (46) has two components which are expressed as follow:

$$\delta U_t + U\delta U_x + V\delta U_y + U_x\delta U + U_y\delta V = Q \star \lambda^1 \tag{62}$$
$$\delta V_t + U\delta V_x + V\delta V_y + V_x\delta U + V_y\delta V = Q \star \lambda^2 \tag{63}$$

Equation (62) is split into:

$$\delta U_t + U\delta U_x + U_x\delta U = 0$$
$$\delta U_t + V\delta U_y + U_y\delta V = Q \star \lambda^1$$

Again, linear advection terms are approximated using shock filter.

$$\frac{\delta U^{k+1}_{i,j} - \delta U^k_{i,j}}{\triangle t} = -S_x(U, \delta U)^k_{i,j} - \frac{1}{2}(U^k_{i+1,j} - U^k_{i-1,j})\delta U^k_{i,j}$$
$$\frac{\delta U^{k+1}_{i,j} - \delta U^k_{i,j}}{\triangle t} = -S_y(V, \delta U)^k_{i,j} - \frac{1}{2}(U^k_{i,j+1} - U^k_{i,j-1})\delta V^k_{i,j} + (Q \star \lambda^1)^k_{i,j}$$

Equation (63) is split into:

$$\delta V_t + U\delta V_x + V_y\delta V = 0$$
$$\delta V_t + V\delta V_y + V_x\delta U = Q \star \lambda^2$$

and approximated by:

$$\frac{\delta V^{k+1}_{i,j} - \delta V^k_{i,j}}{\triangle t} = -S_x(U, \delta V)^k_{i,j} - \frac{1}{2}(V^k_{i,j+1} - V^k_{i,j-1})\delta V^k_{i,j}$$
$$\frac{\delta V^{k+1}_{i,j} - \delta V^k_{i,j}}{\triangle t} = -S_y(V, \delta V)^k_{i,j} - \frac{1}{2}(V^k_{i+1,j} - V^k_{i-1,j})\delta U^k_{i,j} + (Q \star \lambda^2)^k_{i,j}$$

References

1. L. Alvarez, J. Weickert, and J. Sánchez. Reliable estimation of dense optical flow fields with large displacements. International Journal of Computer Vision, 39(1):41–56, 2000.
2. A. Apte, C.K.R.T. Jones, A.M. Stuart, and J. Voss. Data assimilation: Mathematical and statistical perspectives. Int. J. Numer. Meth. Fluids, 56:1033–1046, 2008.
3. T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In Springer-Verlag, editor, Proceedings of European Conference on Computer Vision, volume 4, pages 25–36, Prague, Czech Republic, May 2004.
4. J. Hadamard. Lecture on Cauchy's Problem in Linear Partial Differential Equations. Yale University Press, New Haven, 1923.
5. I. Herlin, F.-X. Le Dimet, E. Huot, and J.-P. Berroir. Coupling models and data: which possibilities for remotely-sensed images? In Poulicos Prastacos, Ulises Cortés, Juan-Luis Díaz De León, and Manuel Murillo, editors, e-Environment: Progress and Challenge, volume 11 of Research on Computing Science, pages 365–383. Instituto Politécnico Nacional, November 2004.
6. B.K.P. Horn and B.G. Schunk. Determining optical flow. Artificial Intelligence, 17:185–203, 1981.
7. E. Huot, I. Herlin, and G. Korotaev. Assimilation of sst satellite images for estimation of ocean circulation velocity. In Proceedings of IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Boston, Massachusetts, U.S.A., July 2008.
8. F. Le-Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects., pages 97–110. Tellus, 1986.
9. F.-X. Le Dimet, I.M. Navon, and D.N. Daescu. Second-order information in data assimilation. Monthly Weather Rev, 130:629–648, March 2002.
10. D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. Communications on Pure and Applied Mathematics, XLII(577–685), 1989.
11. H.-H. Nagel. Displacement vectors derived from second-order intensity variations in image sequences. Computer Vision, Graphics, and Image Processing, 21:85–117, 1983.
12. J.-M. Odobez and P. Bouthemy. Direct incremental model-based image motion segmentation for video analysis. Signal Processing, 66(2):143–155, 1998.
13. D.S. Oliver. Calculation of the inverse of the covariance. Mathematical Geology, 30(7):911–933, 1998.
14. N. Papadakis, T. Corpetti, and É. Mémin. Dynamically consistent optical flow estimation. In Proceedings of International Conference on Computer Vision, Rio de Janeiro, Brazil, October 2007.
15. N. Papadakis, P. Héas, and É. Mémin. Image assimilation for motion estimation of atmospheric layers with shallow-water model. In Proceedings of Asian Conference on Computer Vision, pages 864–874, Tokyo, Japan, November 2007.
16. N. Papadakis and É. Mémin. Variational optimal control technique for the tracking of deformable objects. In Proceedings of International Conference on Computer Vision, Rio de Janeiro, Brazil, October 2007.
17. N. Papadakis, E. Mémin, and F. Cao. A variational approach for object contour tracking. In Proceedings of ICCV'05 Workshop on Variational, Geometric and Level Set Methods in Computer Vision, Beijing, China, October 2005.
18. P. Perona and J. Malik. Space scale and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(7):629–639, 1990.
19. M. Proesmans, L. Van Gool, E. Pauwels, and A. Oosterlinck. Determination of optical flow and its discontinuities using non-linear diffusion. In Proceedings of European Conference on Computer Vision, volume 2, pages 295–304, 1994.
20. J.A. Sethian. Level Set Methods. Cambridge University Press, 1996.
21. A. Tarantola. Inverse Problem Theory and Methods for Model Parameter Estimation. Society for Industrial and Applied Mathematics, 2005.
22. A. N. Tikhonov. Regularization of incorrectly posed problems. Sov. Math. Dokl., 4:1624–1627, 1963.
23. J.G. Verwer and B. Sportisse. A note on operator splitting in a stiff linear case. Technical Report MAS-R9830, Center voor Wiskunde en Informatica, December 1998.
24. J. Weickert. Anisotropic diffusion in image processing. ECMI Series. Teubner-Verlag, Stuttgart, 1998. ISBN:3-519-02606-6.

25. J. Weickert. Applications of nonlinear diffusion in image processing and computer vision. In Acta Math. Univ. Comenianae. Proceeding of Algoritmy 2000, volume LXX, pages 33–50, 2001.
26. J. Weickert and C. Schnörr. Variational optic flow computation with a spatio-temporal smoothness constraint. Journal of Mathematical Imaging and Vision, 14:245–255, 2001.
27. A. P. Witkin. Scale-space filtering. In Proc. 8th Int. Joint Conf. Art. Intell., pages 1019–1022, Karlsruhe, Germany, August 1983.

Using models of dynamics for large displacement estimation on noisy acquisitions

Dominique Béréziat and Isabelle Herlin

***Abstract*—The paper discusses the issue of motion estimation on noisy images displaying large displacements, due to high velocity values. "Noisy" means that the data contain either missing acquisitions on isolated points, regions, frames or noisy measures. Assuming the dynamics is partially accessible from heuristics and modeled, the objective is to include this knowledge in the computation of the solution even if large displacements occur from one frame to the next one and if the data are noisy. This is performed by Data Assimilation techniques which simultaneously solve an evolution equation and an observation equation. The evolution equation includes the partial knowledge on the dynamics. The observation equation describes the transport of image brightness and is written in a non-linear form in order to better characterize large displacements. The assimilation method is a weak 4D-Var algorithm, in which each component of the Data Assimilation system is associated to an error. We prove that the observation covariance matrix can be used to discard the noisy data during the computation of the solution letting the evolution equation estimate motion from adjacent frames on these pixels. The method is quantified on synthetic data and illustrated on oceanographic satellite images.**

***Index Terms*—Optical flow, Variational Data Assimilation, SST images.**

I. Introduction

Motion estimation is one major task of Image Processing. It is applied for instance on satellite data to study the cloud structures dynamics or to estimate the ocean surface circulation. In the medical domain, motion detection is used for analyzing the dynamics of organs: abnormalities of the cardiac cycle are detected with ultrasound imaging and the brain neural activity is controlled with magneto and electroencephalography (MEG/EEG) or magnetic resonance imaging (MRI). In video surveillance, motion information permits to assess people behavior.

Motion vectors fields are often inferred from the Optical Flow Constraint [5] but this equation is only valid for small displacements as it results from the linearization of a non linear equation. However, large displacements frequently occur if the time period of the acquisition process is too large compared to the observed dynamics. There are two main approaches to deal with this issue. The first one solves the Optical Flow Constraint equation with an incremental algorithm [8], [12], each iteration remaining a linear problem. The second one directly solves the non linear equation of image brightness transport. Such method is named warping method [4].

An additional difficulty concerns the quality of images: as widely known, the image data are commonly noisy. This degradation has various origins: a failing acquisition process, occluding structures (clouds over the sea, people behind objects, ...), or a poor quality of the signal (ultrasound imaging for instance). The corresponding values should be ignored or given a low weight during the estimation process and the final result.

Estimating a motion field from two consecutive images with the optical flow constraint is an ill-posed problem and additional information must be provided to obtain a unique solution. Usually the uniqueness is obtained by constraining the space of the solutions. This method, named Tikhonov regularization, is widely used in the literature [13]. However, this is too restrictive and not always justified according to the experimental context.

Sometimes the underlying dynamics is partially known. This is the case if the physical processes, visualized on images, are accurately modeled by the specialists. In meteorology and oceanography, equations, approximately describing apparent motion on the image data, can be derived from the fluid flow dynamics. In the medical domain, efforts are engaged to describe the electro-mechanic cardiac cycle and the brain electric activity, and first dynamic models are now available. The problem is then to include this knowledge in the computation of the solution: this is achieved using Data Assimilation methods. Knowing the approximate dynamics provides two major advantages as it had been discussed in [2]: first, a unique solution can be obtained without any Tikhonov regularisation and, second, a solution is obtained even if the observation are noisy. Using Data Assimilation to determine motion from image sequences is an emerging domain. In [10], the optical flow constraint is used as observation equation to compute velocity and two alternative evolution equations are considered: a Navier-Stokes equation ruling the fluid velocity and a transport of velocity by itself for addressing rigid motions. In [11], the velocity field of several atmospheric layers is computed from pressure images and shallow-water equations are used as evolution equation for velocity and pressure. In [6], the surface velocity field is estimated from SST images with a shallow-water model and an advection/diffusion equation for the temperature.

In this paper, the issue of motion estimation is investigated in the following context: the displacements may be large and approximately ruled by the transport of brightness; observations may be noisy but the localization of noisy pixels is known by the user from metadata. Without any information on the image dynamics, the transport of velocity by itself is considered. Such assumption means that the motion vector is constant over time along the pixel trajectory. As this model is rough, derivation from it is accepted in the estimation process. In Section II, the issue of motion estimation is introduced, and equations ruling the evolution of brightness

and the dynamics are discussed. In Section III, the variational data assimilation framework is briefly described for a complete understanding of our method. Section IV gives a discussion about the covariance matrix associated to the error involved in the observation equation. We explain how to use it to discard noisy data from the solution and to add spatio-temporal regularity constraints on the solution. In Section V, results are quantified on synthetic images and displayed on SST data acquired over the Black Sea. Concluding remarks and perspectives are given in Section VI.

II. Defining the evolution over the image sequence

Optical flow measures the displacement of patterns on a sequence of images. This is one signature of the physical processes occurring in the scene during the acquisition. Equations ruling it therefore depend on the studied context. A first example concerns the estimation of fluid particles velocity from a mass conservation equation:

$$\frac{\partial I}{\partial t} + \nabla^T (I\mathbf{V}) = 0 \tag{1}$$

with I denoting the image value, ∇ the spatial gradient, ∇^T its transpose and $\mathbf{V}$ the velocity. This equation is suitable if, for instance, I is an image of pressure and $\mathbf{V}$ the blood speed [15] or I is an image of temperature (satellite infrared image) and $\mathbf{V}$ the clouds displacement [3], [7]. A second example, commonly used for video sequences, assumes that moving objects display a Lambertian surface and the optical flow is modeled as the transport of image brightness by the velocity vector:

$$I(\mathbf{x} + \mathbf{V}(\mathbf{x};t)\triangle t; t + \triangle t) = I(\mathbf{x};t)$$

$\mathbf{x}$ and t are space-time coordinates, $\triangle t$ is the time step between two frames and can be set to 1 without any loss of generality:

$$I(\mathbf{x} + \mathbf{V}(\mathbf{x};t); t+1) = I(\mathbf{x};t) \tag{2}$$

Being non linear, Equation (2) is commonly approximated by a first-order Taylor expansion:

$$\frac{\partial I}{\partial t} + \nabla I^T \mathbf{V} = 0 \tag{3}$$

Equations (1), (2) and (3) are named *observation equations* as they link the observations and the quantity to be computed, named the state vector, which is the velocity field $\mathbf{V}$.

As we are interested in tracking structures visualized on the ocean surface, the displacements may be large if the time period between two frames is high compared to the dynamics time scale. Equation (3) is then no more valid and Equation (2) is considered. This observation equation is not invertible and provides an infinity of solutions. A first possibility to overcome this difficulty is to look for the velocity $\mathbf{V}$ as the minimum of an objective function and to include a constraint on the velocity spatial variation. This is named the Tikhonov regularization [13]:

$$\int \left((I(\mathbf{x} + \mathbf{V}; t+1) - I(\mathbf{x};t))^2 + \alpha^2 \|\nabla \mathbf{V}\|^2 \right) d\mathbf{x} \tag{4}$$

The associated Euler-Lagrange equation is established and leads to a non linear system. A semi-implicit scheme [4] can then be used with the drawback of requiring the inversion of a large matrix. A second possibility is to consider an incremental method such as in [12]. In this case, we consider the value $I(\mathbf{x} + \tilde{\mathbf{V}}; t+1)$ with $\tilde{\mathbf{V}}$ being the current estimate of the velocity $\mathbf{V}$, and subtract it in Equation (2). We obtain after a first-order Taylor expansion:

$$\nabla I^T(\mathbf{x};t)(\mathbf{V} - \tilde{\mathbf{V}}) + I(\mathbf{x} + \tilde{\mathbf{V}}; t+1) - I(\mathbf{x};t) = 0 \tag{5}$$

This equation is also ill-posed and requires a Tikhonov regularization. The new objective function is:

$$\int \left(\nabla I^T(\mathbf{V} - \tilde{\mathbf{V}}) + I(\mathbf{x} + \tilde{\mathbf{V}}; t+1) - I)^2 + \alpha^2 \|\nabla \mathbf{V}\|^2 \right) d\mathbf{x} \tag{6}$$

Equation (5) being linear, the Euler-Lagrange equation associated to (6) is linear and can be solved efficiently with a Gauss-Seidel method [5]. It provides a solution for $\mathbf{V}$ from the estimate $\tilde{\mathbf{V}}$. Due to the Taylor expansion, the equation provides an approximate solution of the transport equation (2). The process is then iterated until convergence using the solution from an iteration as the initial estimate for the next one.

However the dynamics of $\mathbf{V}$ may be sometimes approximated from heuristics and described by an evolution equation:

$$\frac{\partial \mathbf{V}}{\partial t} + \mathbb{M}(\mathbf{V}) = 0 \tag{7}$$

The operator $\mathbb{M}$ is named *evolution model*. Given an initial condition for $\mathbf{V}$ at time $t = 0$, it is possible to compute $\mathbf{V}$ at any time $t > 0$ by integrating (7). The Tikhonov regularization, corresponding to the second term of (6), is no more required.

Depending on the experimental context, various evolution models may be considered. The best would be to consider the mathematical equations describing the observed physical processes. For instance in oceanography, the heat equation rules the diffusion of the surface temperature and the shallow-water equations describe the ocean surface circulation [6]. These shallow-water equations are a simplification of the Navier-Stokes equation describing turbulent and incompressible fluid flows:

$$\rho \frac{d\mathbf{V}}{dt} = \mathbf{F} - \nabla p + \mu \nabla^2 \mathbf{V} \tag{8}$$

$\mathbf{F}$ being the gravity force, p the pressure, ρ the fluid density, and μ the kinetic viscosity. Without effective knowledge on dynamics, simpler equations can be experimented, as the Lagrangian constancy:

$$\frac{d\mathbf{V}}{dt} = \frac{\partial \mathbf{V}}{\partial t} + (\mathbf{V}^T \nabla)\mathbf{V} = 0, \tag{9}$$

However these evolution models are often an approximation of the reality and deviations to them must be allowed during the computation, as it is the case with weak data assimilation techniques. Equation (9) is considered in the remainder of the paper.

III. Data Assimilation

Let $\mathbf{X}$ be the state vector and $\mathbf{Y}$ the observation vector defined on a spatio-temporal domain $A = \Omega \times [0,T]$. Data Assimilation solves the following system of three equations:

$$\frac{\partial \mathbf{X}}{\partial t}(\mathbf{x},t) + \mathbb{M}(\mathbf{X})(\mathbf{x},t) = \mathcal{E}_m(\mathbf{x},t) \quad (11)$$

$$\mathbb{H}(\mathbf{Y},\mathbf{X})(\mathbf{x},t) = \mathcal{E}_o(\mathbf{x},t) \quad (12)$$

$$\mathbf{X}(\mathbf{x},0) = \mathbf{X}_b(\mathbf{x}) + \mathcal{E}_b(\mathbf{x}) \quad (13)$$

In addition to the evolution and observation equations (Eqs. (11,12)), a constraint on the initial condition, Eq. (13), is added. For linking with the previous section, Eq. (11) and Eq. (12) respectively correspond to Eq. (7) and Eq. (3) with $\mathbf{X} = \mathbf{V}$ and $\mathbf{Y} = I$ in the case of motion estimation from image data. An error term is added to each equation: $\mathcal{E}_m$ represents the deviation to the evolution equation, $\mathcal{E}_o$ describes the errors on observation data and the deviation from the observation equation, and $\mathcal{E}_b$ concerns the initial condition uncertainty. These errors are supposed unbiased and Gaussian, and fully characterized by their covariance matrix Q, R and B. For solving the full system, the functional (10) has to be minimized with respect to $\mathbf{X}$. Its differential is established by determining the directional derivative and introducing the auxiliary variable (named *adjoint variable*) $\lambda(\mathbf{x},t) = \int_A Q^{-1}(\mathbf{x},t,\mathbf{x}',t') \left(\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X}) \right) (\mathbf{x}',t') d\mathbf{x}' dt'$. As operators $\mathbb{M}$ and $\mathbb{H}$ are non linear, a local linearization is applied: the state vector is written as $\mathbf{X} = \mathbf{X}_b + \delta\mathbf{X}$, with $\mathbf{X}_b$ the *background variable* and $\delta\mathbf{X}$ the *incremental variable*. A Taylor development of $\mathbb{M}$ and $\mathbb{H}$ is performed at the neighborhood of $\mathbf{X}_b$. The following system of equations has then to be solved (reader is referred to [2], [14] for a complete description):

$$\lambda(\mathbf{x},\mathbf{T}) = 0 \quad (14)$$

$$-\frac{\partial \lambda}{\partial t} + \left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b} \right)^* \lambda = -\int_A \left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b} \right)^* R^{-1} \left(\mathbb{H}(\mathbf{X}_b,\mathbf{Y}) + \frac{\partial \mathbb{H}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b} (\delta\mathbf{X}) \right) d\mathbf{x}' dt' \quad (15)$$

$$\mathbf{X}_b(\mathbf{x},0) = \mathbf{X}_b(\mathbf{x}) \quad (16)$$

$$\frac{\partial \mathbf{X}_b}{\partial t} + \mathbb{M}(\mathbf{X}_b) = 0 \quad (17)$$

$$\delta\mathbf{X}(\mathbf{x},0) = \int_\Omega B\lambda(\mathbf{x}',0) d\mathbf{x}' \quad (18)$$

$$\frac{\partial \delta\mathbf{X}}{\partial t} + \frac{\partial \mathbb{M}}{\partial \mathbf{X}}\Big|_{\mathbf{X}_b} (\delta\mathbf{X}) = \int_A Q\lambda(\mathbf{x}',t') d\mathbf{x}' dt' \quad (19)$$

Operators $\left(\frac{\partial \mathbb{M}}{\partial \mathbf{X}} \right)^*$ and $\left(\frac{\partial \mathbb{H}}{\partial \mathbf{X}} \right)^*$ are the *adjoint operators*. For a given operator $\mathbb{K}$, the adjoint is mathematically defined by:

$$\langle \mathbb{K}(\eta)), \lambda \rangle = \langle \eta, \mathbb{K}^*(\lambda) \rangle \quad (20)$$

The linearization of operators $\mathbb{M}$ and $\mathbb{H}$ leads to an inexact solution of Equations (14) to (19) and requires to iterate this incremental method. The background variable is first computed from Equations (16) and (17). Background and incremental variables are used to compute the adjoint variable with Equations (14) and (15). After updating the background variable with $\mathbf{X}_b = \mathbf{X}_b + \delta\mathbf{X}$, a new value for the incremental variable can then be computed from Equations (18) and (19). The process is iterated up to convergence. This method, named incremental 4D-var, has been designed specifically to solve non linear evolution and observation models such as those discussed in the previous section.

IV. The error covariance matrix

This section describes the use of error covariance matrices to handle noisy data and to express spatio-temporal regularity properties.

First, for a general discussion, let Z be a stochastic state vector depending on the spatial coordinate $\mathbf{x}$ and Σ its covariance matrix:

$$\Sigma(\mathbf{x},\mathbf{x}') = \iint (Z(\mathbf{x}) - \mathbb{E}(Z))^T (Z(\mathbf{x}') - \mathbb{E}(Z)) dP(\mathbf{x},\mathbf{x}')$$

$dP(\mathbf{x},\mathbf{x}')$ being the density of $(Z(\mathbf{x}), Z(\mathbf{x}'))$ and $\mathbb{E}$ the expectation. The inverse of Σ is defined in [9] by:

$$\int \Sigma^{-1}(\mathbf{x},\mathbf{x}'')\Sigma(\mathbf{x}'',\mathbf{x}')d\mathbf{x}'' = \delta(\mathbf{x}-\mathbf{x}') \quad (21)$$

A first class of covariance matrices concerns the one depending on $\mathbf{x}-\mathbf{x}'$: $\Sigma(\mathbf{x},\mathbf{x})' = \Sigma(\mathbf{x}-\mathbf{x}')$. Equation (21) can be rewritten as a convolution product $\int \Sigma^{-1}(r-r')\Sigma(r')dr' = \delta(r)$ with $r = \mathbf{x}-\mathbf{x}'$ and $r' = \mathbf{x}''-\mathbf{x}'$. This corresponds to a simple product $\widehat{\Sigma}^{-1}(\omega)\widehat{\Sigma}(\omega) = 1$ in the Fourier space. It is then much easier to compute the inverse covariance matrix in the Fourier domain. A Dirac covariance, $\Sigma(\mathbf{x},\mathbf{x}') = \delta(\mathbf{x}-\mathbf{x}')$, has an inverse which is also a Dirac function, and we have $\iint Z^T(\mathbf{x})\Sigma^{-1}Z(\mathbf{x}')d\mathbf{x}d\mathbf{x}' = \int \|Z(\mathbf{x})\|^2 d\mathbf{x}$. Applying a Dirac covariance is therefore equivalent to a zero-order regularization. For an exponential covariance, $\Sigma(\mathbf{x},\mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|}{\sigma}\right)$, we have $\iint Z^T(\mathbf{x})\Sigma^{-1}Z(\mathbf{x}')d\mathbf{x}d\mathbf{x}' = \int \frac{1}{2\sigma}\left(\|Z(\mathbf{x})\|^2 + \sigma^2\|\nabla Z(\mathbf{x})\|^2\right) d\mathbf{x}$ [2], which is equivalent

$$\begin{aligned} E(\mathbf{X}) = &\int_A\int_A \big(\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X})\big)^T(\mathbf{x},t)Q^{-1}(\mathbf{x},t,\mathbf{x}',t')\big(\frac{\partial \mathbf{X}}{\partial t} + \mathbb{M}(\mathbf{X})\big)(\mathbf{x}',t')d\mathbf{x}dtd\mathbf{x}'dt' \\ &+ \int_A\int_A \mathbb{H}(\mathbf{X},\mathbf{Y})^T(\mathbf{x},t)R^{-1}(\mathbf{x},t,\mathbf{x}',t')\mathbb{H}(\mathbf{X},\mathbf{Y})(\mathbf{x}',t')d\mathbf{x}dtd\mathbf{x}'dt' \\ &+ \int_\Omega\int_\Omega \big(\mathbf{X}(\mathbf{x},0) - \mathbf{X}_b(\mathbf{x})\big)^T B^{-1}(\mathbf{x},\mathbf{x}')\big(\mathbf{X}(\mathbf{x}',0) - \mathbf{X}_b(\mathbf{x}')\big)d\mathbf{x}d\mathbf{x}' \end{aligned} \quad (10)$$

to a first-order regularization.

A second class concerns covariance matrices written as $\Sigma(\mathbf{x},\mathbf{x}') = S(\mathbf{x})\delta(\mathbf{x}-\mathbf{x}')$. We have $\Sigma^{-1}(\mathbf{x},\mathbf{x}') = \delta(\mathbf{x}-\mathbf{x}')S^{-1}(\mathbf{x})$. In this case, the error is localized. These matrices are used to weight the contribution of pixels in the energy function (10).

To avoid noisy data having an impact on the solution, we define the inverse of the observation covariance matrix R as:

$$R^{-1}(\mathbf{x},t;\mathbf{x}',t') = \delta(\mathbf{x}-\mathbf{x}')\delta(t-t')f(\mathbf{x},t) \qquad (22)$$

with f a confidence measure which is close to 0 if the observation value must be discarded because the acquisition process failed or the observation model is wrong. A first category of pixels is identified from metadata with a function f_{sensor} that expresses the quality of the acquisition. For example SST images contain clouds occluding the ocean surface: these areas are flagged by the provider and f_{sensor} has a low value on these pixels. A second category concerns pixels having a small value of the space-time gradient norm: Equation (2) becomes true whatever the value of $\mathbf{V}$ as it reduces to $0 = 0$. To discard these pixels from the computation we define the quality function $f_{\mathbb{H}}$ such as:

$$f_{\mathbb{H}}(\mathbf{x},t) = 1 - \exp\left(-\frac{\|\nabla_3 I(\mathbf{x},t)\|^2}{\sigma}\right) \qquad (23)$$

with ∇_3 the gradient operator on the space-time domain. The function f in Eq. (22) is then defined by $f(\mathbf{x},t) = f_{\text{sensor}}(\mathbf{x},t)f_{\mathbb{H}}(\mathbf{x},t)$.

Another concern is the spatial regularization of the result. Our objective is to perform this regularization with the observation error covariance matrix R. Applying a spatial regularization to the state vector $\mathbf{X}$ may be seen as defining a piecewise constant approximation $\mathbf{X}_{\text{pc}}$. This issue is formalized by extending the observation vector as $\mathbf{Y}' = (\mathbf{Y},0)^T$: the observations $\mathbf{Y}$ will be compared to $\mathbf{X}$ as previously and null observations of $\nabla\mathbf{X}$ are given in order to force $\mathbf{X}$ to be piecewise constant. For the optical flow estimation, the observation equation is rewritten as:

$$\mathbb{H}'(\mathbf{V},I) = \begin{pmatrix} I(\mathbf{x}+\mathbf{V},t+1) - I(\mathbf{x},t) \\ \nabla\mathbf{V} \end{pmatrix} = \mathcal{E}_o \qquad (24)$$

This is a two-component equation and the inverse of the error covariance matrix is defined as:

$$R'(\mathbf{x},t;\mathbf{x}',t') = \begin{pmatrix} R^{-1}(\mathbf{x},t;\mathbf{x}',t') & 0 \\ 0 & \gamma\delta(\mathbf{x}-\mathbf{x}') \end{pmatrix} \qquad (25)$$

with $\gamma > 0$ and R^{-1} written as in Eq. (22). Using $\mathbb{H}'$ and R'^{-1} from Eq. (24) and (25), with $\gamma = \alpha^2$, we have:

$$\iint \mathbb{H}'(\mathbf{X},\mathbf{Y}')^T R'^{-1}\mathbb{H}'(\mathbf{X},\mathbf{Y}')d\mathbf{x}dtd\mathbf{x}'dt' =$$
$$\int \left(f(\mathbf{x},t)(I(\mathbf{x}+\mathbf{V},t+1) - I(\mathbf{x},t))^2 + \alpha^2\|\nabla\mathbf{V}\|^2\right)d\mathbf{x}dt \qquad (26)$$

The second part of this integral may be recognized as the classical Tikhonov regularization involved in Eq. (4). The data assimilation method, applied on the state vector $\mathbf{X}$, with the observation $\mathbf{Y}'$ and the matrix R', actually performs the same regularization in an indirect way.

V. Implementation and Results

Details for determining the differential and adjoint operator of $\mathbb{M}(\mathbf{V}) = \nabla\mathbf{V}^T\mathbf{V}$ are given in [2]. The differential of $\mathbb{H}(\mathbf{V},I) = I(\mathbf{x}+\mathbf{V},t+1) - I(\mathbf{x},t)$ is $\nabla I^T(\mathbf{x}+\mathbf{V},t+1) \overset{\text{def}}{=} \nabla I^T \circ \mathbf{V}$ [4] and the adjoint operator is the transpose. A Dirac covariance is chosen for Q and B. Equations (17,15,19) correspond to the three following EDPs:

$$\frac{\partial\mathbf{V}}{\partial t} + \mathbf{V}_b^T\nabla\mathbf{V}_b = 0 \qquad (27)$$

$$-\frac{\partial\lambda}{\partial t} - \nabla\lambda^T\mathbf{V}_b - \lambda^T\nabla^\perp\mathbf{V}_b = L \qquad (28)$$

$$\frac{\partial\delta\mathbf{V}}{\partial t} + \mathbf{V}_b^T\nabla\delta\mathbf{V} + \nabla\mathbf{V}_b^T\delta\mathbf{V} = \lambda \qquad (29)$$

The right member of Equation (28) is:

$$L = -\nabla I \circ \mathbf{V}_b\left(I\circ\mathbf{V}_b - I + \nabla I^T\circ\mathbf{V}_b\delta\mathbf{V}\right)f_{\mathbb{H}}f_{\text{sensor}}$$

Discretisation is performed as follows. Equation (27) is a non linear advection and a robust explicit numerical scheme is proposed in [2]. Equations (28) and (29) have linear but non constant advection terms with additional forcing terms. The advection terms are discretized using an upwind scheme [1].

The initial condition $\mathbf{X}_b$, in Equation (16), is given by the Horn & Schunck algorithm [5].

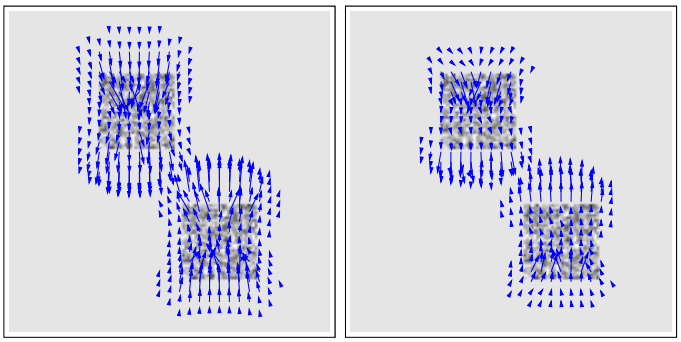
A. Synthetic data

The potential of the method is first demonstrated on a synthetic sequence displaying two squares, one moving down and the other moving up, meeting each other on the last frames. Figures 1(a) and 1(b) show the results obtained at the beginning and the end of the sequence for Horn & Schunck (HS) and Data Assimilation (DA) methods. HS has the drawback of spatially smoothing the velocity field at the junction of the two squares at the end of the sequence producing sometimes a wrong direction for the motion vector. The heuristics on the dynamics is successfully exploited by DA that estimates a correct direction of the motion field.

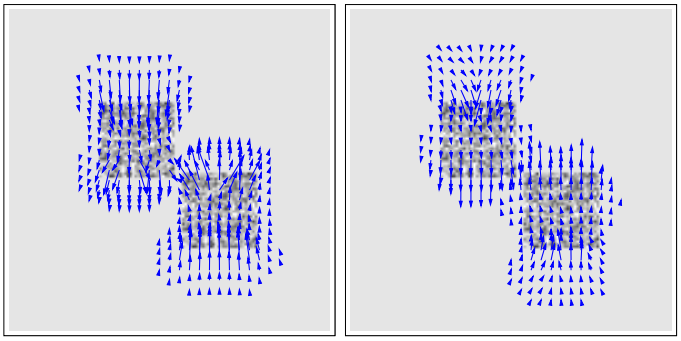
B. Satellite data

The algorithm has also been used on a sequence of 17 SST images, acquired by NOAA-AVHRR in July 1998 over the Black Sea. The image data present major difficulties: (1) Figure 2(a) and 2(b) display two consecutive frames with a cloud occluding a large region and displaying a large displacement; (2) Figure 2(c) shows data missing due to a registration problem; (3) the pixel brightness is subject to large variations on the sequence due to varying acquisition times and atmospheric conditions (see frame 10 in Figure 2(d) which is darker than other frames). To deal with this last problem, we use the coarse hypothesis of constant brightness variation along the trajectory: $\frac{dI}{dt} = a$, with a being a real constant. Consequently $\frac{\partial}{\partial x}\left(\frac{dI}{dt}\right) = \frac{d}{dt}\left(\frac{\partial I}{\partial t}\right) = 0$ and we consider $\frac{\partial I}{\partial x}$ as input data instead of the image function I.

For analyzing results, we focus on a small region of interest, marked by the blue square in Figure 2(b). Magnification is displayed at several acquisition dates in Figure 2(d). It shows a black structure evolving with a translational motion,



(a) Result on frame 3 for HS(left) DA (right).



(b) Result on frame 9 for HS (left) and DA (right).

Fig. 1. Results on synthetic data.

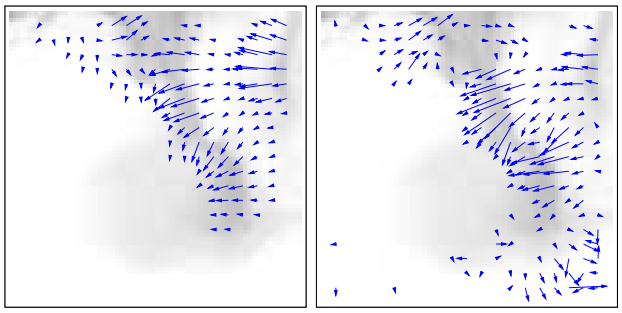
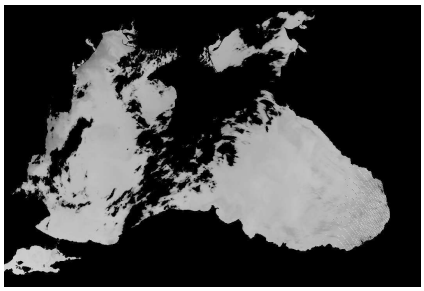


Fig. 3. Result on SST with HS (left) and DA (right) on frame 2.

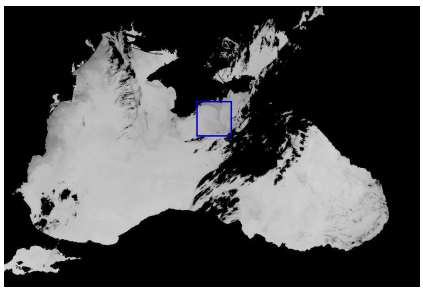
from right to left, during the first frames and then with a counter clockwise rotational motion during the remainder of the sequence. The acquisitions also display missing data on frames 9 and 11. Figure 3 shows the results of HS and DA on the second frame. The two methods give similar results. This demonstrates that the spatial regularity involved in HS may be suppressed and replaced by the evolution equation if the dynamics is correctly modeled. Figure 4 displays the results obtained on the third and fourth frames (partially occluded by a cloud). As HS computes spatio-temporal gradients using consecutive frames, it fails if gradients are computed on missing data. DA, however, provides a good result because these pixels are discarded due to the confidence function f: the solution is obtained from the evolution equation. Figure 5 illustrates results on the tenth and eleventh frames with many missing pixels. HS fails to provide a coherent velocity field. DA successes again thanks to the evolution equation and the discard of missing data.

VI. Concluding remarks

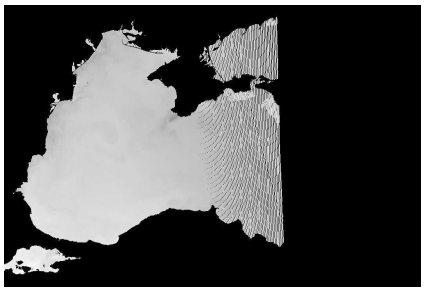
In the paper, we proposed a method to estimate large displacements from an image sequence including some noisy data. The method relies on an evolution equation which approximately describes the image dynamics. Data assimilation is used to simultaneously solve the evolution and brightness transport equations. Noisy data are identified by a confidence function and discarded from the computation of the solution, that is then estimated from the evolution equation on these pixels. A spatial regularization of the solution may be obtained from a suitable observation equation and its covariance matrix.



(a) Large cloud in black occluding the Sea.



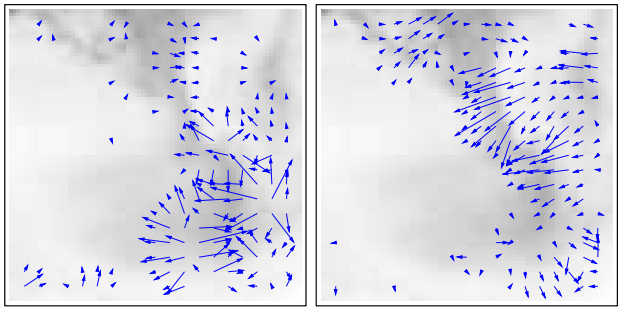
(b) Next frame.



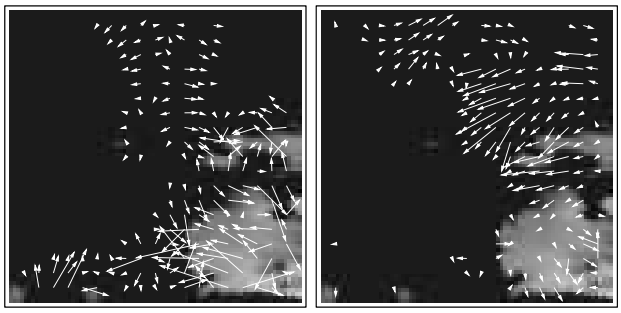
(c) Missing data on the right part of the acquisition.

(d) Region of interest on frames 2, 3, 4, 9, 10 and 11.

Fig. 2. SST sequence with noisy data.

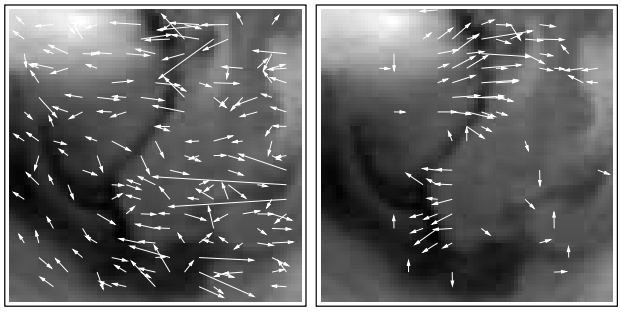


(a) Frame 3.

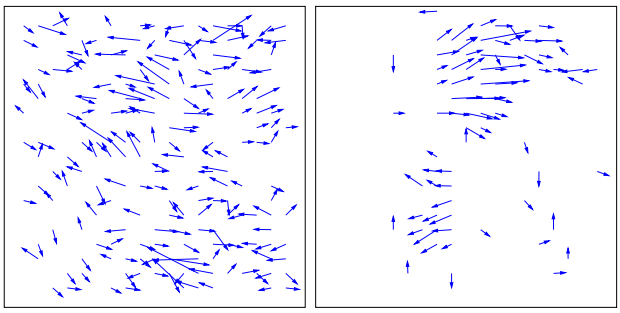


(b) Frame 4.

Fig. 4. Cloud occultation: HS (left) and DA (right).



(a) Frame 10.



(b) Frame 11. The background – see frame 11 in Figure 2(d) – has been removed in order to enhance visibility.

Fig. 5. Missing data: HS (left) and DA (right).

We discussed the relevance of the method on satellite images displaying various types of missing data. As the dynamics is obtained from heuristics, we allowed inaccuracies of the evolution equation by involving a model error in the equations. Even if the dynamics' model is inaccurate, we obtain better results than standard optical flow methods because the missing data are correctly managed by our method.

The major perspective is to define an evolution model dealing with spatio-temporal discontinuities of the dynamics. If the state vector is too different from the observations at a given time step, the method should consider a dynamics' discontinuity and the evolution equation should no more be involved in the computation of the solution. This could be obtained by increasing the error model and its covariance matrix Q. However the problem will then become ill-posed and require a spatial regularization at that time step. The difficulty is to define adaptive covariances Q and R driven by the observations and the state vector in order to release the weight of the evolution equation and to apply regularization if necessary.

REFERENCES

[1] W. Ames. *Numerical Methods for Partial Differential Equations*. Academic Press, 1977. 4

[2] D. Béréziat and I. Herlin. Solving ill-posed image processing problems using data assimilation. *Numerical Algorithms*, to appear, 2010. 1, 3, 4

[3] D. Béréziat, I. Herlin, and L. Younes. A generalized optical flow constraint and its physical interpretation. In *CVPR*, pages 487–492, June 2000. 2

[4] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, volume 4, pages 25–36, May 2004. 1, 2, 4

[5] B. Horn and B. Schunk. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981. 1, 2, 4

[6] E. Huot, I. Herlin, and G. Korotaev. Assimilation of SST satellite images for estimation of ocean circulation velocity. In *IGARSS*, July 2008. 1, 2

[7] T. Isambert, J. Berroir, and I. Herlin. A multiscale vector spline method for estimating the fluids motion on satellite images. In *ECCV*. Springer, Oct. 2008. 2

[8] J.-M. Odobez and P. Bouthemy. Direct incremental model-based image motion segmentation for video analysis. *Signal Processing*, 66(2):143–155, 1998. 1

[9] D. Oliver. Calculation of the inverse of the covariance. *Mathematical Geology*, 30(7):911–933, 1998. 3

[10] N. Papadakis, T. Corpetti, and E. Mémin. Dynamically consistent optical flow estimation. In *ICCV*, Oct. 2007. 1

[11] N. Papadakis, P. Héas, and E. Mémin. Image assimilation for motion estimation of atmospheric layers with shallow-water model. In *ACCV*, pages 864–874, Nov. 2007. 1

[12] M. Proesmans, L. Van Gool, E. Pauwels, and A. Oosterlinck. Determination of optical flow and its discontinuities using non-linear diffusion. In *ECCV*, volume 2, pages 295–304, 1994. 1, 2

[13] A. N. Tikhonov. Regularization of incorrectly posed problems. *Sov. Math. Dokl.*, 4:1624–1627, 1963. 1, 2

[14] E. Valur Hólm. Lectures notes on assimilation algorithms. Technical report, European Centre for Medium-Range Weather Forecasts Reading, June 2003. 3

[15] R. Wildes and M. Amabile. Physically based fluid flow recovery from image sequences. In *CVPR*, pages 969–975, June 1997. 2

A.5 Occupation des sols

[1] R. Teina, D. Béréziat, and B. Stoll. A spatial poisson process to classify coconut fields on Ikonos pansharpened images. In *Multispectral, Hyperspectral, and Ultraspectral Remote Sensing Technology, Techniques, and Applications II. Proceedings of SPIE*, volume 7149, page 71491E (10 pages), Noumea (New Caledonia), November 2008.

[2] R. Teina, D. Béréziat, B. Stoll, and S. Chabrier. Toward a global Tuamotu Archipelago trees sensing using high resolution optical data. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, volume II, Boston (USA), July 2008.

[3] R. Teina, D. Béréziat, B. Stoll, and S. Chabrier. A comparative study of several supervised classifiers for coconut tree field's type mapping on 80 cm rgb pansharpened Ikonos images. In *Image Processing : Machine Vision Applications II. Proceedings of SPIE*, volume 7251, page 72510X (9 pages), San Jose (CA), January 2009.

A Spatial Poisson Point Process to classify coconut fields on Ikonos pansharpened images

Teina R.[a] and Béréziat D.[a] and Stoll B.[b]

[a]Université Pierre et Marie Curie, LIP6, 104 avenue du Président Kennedy, 75016 Paris, France;
[b]Université de la Polynésie Française, Laboratoire GePaSud, 98702 Faa'a, Tahiti - French Polynesia;

ABSTRACT

The goal of this study is to classify the coconut fields, observed on remote sensing images, according to their spatial distribution. For that purpose, we use a technique of point pattern analysis to characterize spatially a set of points. These points are obtained after a coconut trees segmentation process on Ikonos images. Coconuts' fields not following a Poisson Point Process are identified as maintained, otherwise other fields are characterized as wild. A spatial analysis is then used to establish locally the Poisson intensity and therefore to characterize the degree of wildness.

Keywords: Poisson Point Process, Ripley's function, quadrant count, nearest neighbor distances, coconut fields typology.

1. INTRODUCTION

French Polynesia government wants to improve their knowledge about the coconut (*Cocos Nucifera L.*) fields and specially to evaluate the coconut fields regeneration of old trees planted in the 1980's. The goal is to enhance the management of Coprah oil extraction as an alternative fuel. The French Polynesia Government acquired in 2003 some RGB pansharpened Ikonos images at 1m pixel resolution on ground for this purpose. Coconuts are detected using a segmentation process based on a marker-controlled watershed algorithm. For each crown, a center of mass, weighted by the values of the pixels in each RGB channels, is computed to estimate the canopies' center.[1] We finally obtain a set of point that we want to characterize as several type according to their spatial distribution.

In this study, we use Poisson Point Process (PPP) to characterize wild (or natural) coconut plantations. If the point pattern does not follow a Poisson Process Point, then the planting is considered as maintained. If the plantations are natural, and therefore identified as a Poisson Point Pattern, then we seek to segment these regions according to a criterion of homogeneity based on the density of the coconut trees. This density is directly related to the parameter of the Poisson distribution. If the density is not uniform, it has to be estimated locally.

We first focus on the Quadrant Count Method. It can be described simply by partitioning the data set into equal sized sub regions. Then some statistics are computed in each quadrant to determine if the observed data are clustered or not.

Another method is to simulate different outcomes of a Poisson Point Pattern, whose the intensity has been estimated from the observation, to compare these achievements with the observation using a suitable distance such as Ripley's K-function,[2,3] L-function, Pair Correlation function (g-function), and to deduce whether the point pattern is spatially uniform or not.

In this paper, we propose to segment regions by analysis of the histogram of the estimated density and according to some theoretical density values for this specific species of coconut trees. We then obtain three regions

Further author information: (Send correspondence to Teina R.)
Teina R.: E-mail: Raimana.Teina@lip6.fr, Telephone: +33 (0) 1 44 27 88 16
Béréziat D.: E-mail: Dominique.Bereziat@lip6.fr

that we can classify as less wild, maintained and wild or natural plantations with the result of a comparison to a Poisson process.

These point pattern analysis methods are applied to coconut field types on the island of Tikehau.

2. DATA AND STUDY AREAS

The study focuses on the atoll of Tikehau, which is located in the north-east of the French Polynesia Territory, specially on the main *motu* (a *motu* is one of the small islands constituting an atoll).
Previous works were done to segment the trees' crown using a marker-controlled watershed approach.[1] Thus, for each crown, a weighted center of mass is computed using informations from the three channel R,G and B of the Ikonos image. Each tree's center is then located by its Cartesian coordinates. Figure 1 shows a picture of the atoll of Tikehau and the main *motu* on which six study regions are located. These regions are representative of the several fields type available on the atoll.

Figure 1. The Atoll of Tikehau and the study areas on the main *motu*.

3. THEORETICAL BACKGROUND

3.1 Spatial Poisson Point Process

A Poisson Point Process is a set of points $X = \{X_i\}$ in a domain $S \subset \mathbb{R}^n$ verifying the following properties:

1. For any $B \subset S$, the number of points belonging to B, denoted $|X \cap B|$ in the remain of this paper, follows a Poisson law with parameter $\lambda\mathcal{A}(B)$, i.e.

$$P(|X \cap B| = n) = e^{-\lambda\mathcal{A}(B)} \frac{(\lambda\mathcal{A}(B))^n}{n!} \tag{1}$$

   where $\mathcal{A}(B)$ stands for the area of B.

2. If B_1 and B_2, two subsets of S, are disjoint, then $|X \cap B_1|$ and $|X \cap B_2|$ are independent.

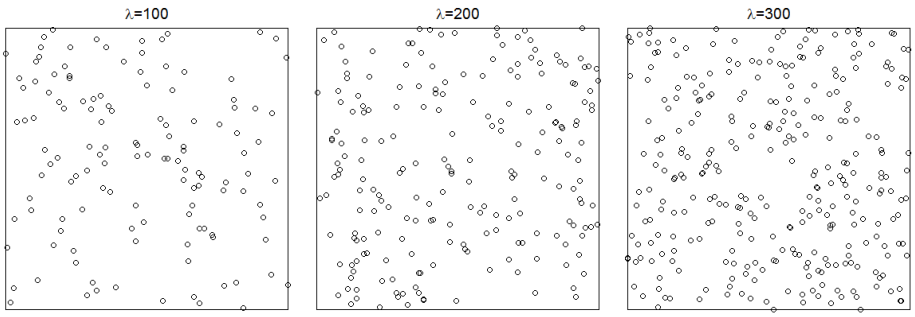
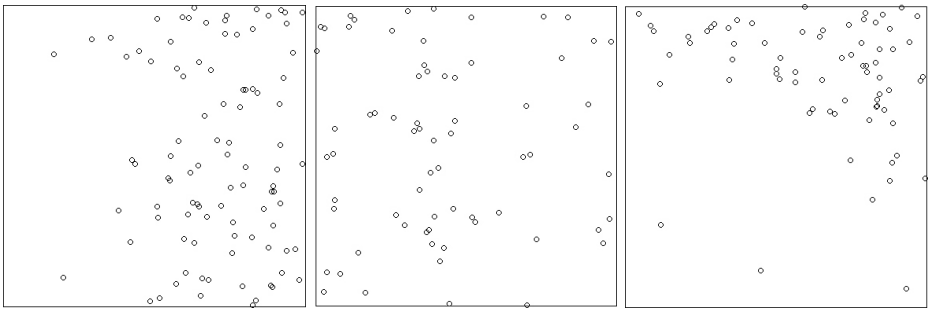


Figure 2. Uniform Poisson point process simulated in the unit square with $\lambda = 100, 200$ and 300.



(a) $\lambda(x, y) = e^{2+5x}$ (b) $\lambda(x, y) = e^{-2x}$ (c) $\lambda(x, y) = e^{2x+5y}$

Figure 3. Inhomogeneous Poisson point process simulated in the unit square using different intensity function $\lambda(x, y)$.

The parameter λ, which fully characterizes the Poisson process, is called intensity. Often, this parameter depends on a spatial coordinate, because the distribution may differ from a region to another one. Such a process is then called Spatial Poisson Point Process (SPPP). A SPPP can be used to characterize, for example, a random distribution of points in space, like a wild coconut field, by building a statistical test: a set of points coordinates are distributed as a Poisson process ($\mathcal{H}_0$) against a non Poisson process (which can characterize a domestic coconut field for instance).

One possible way to characterize a set of point pattern as SPPP is to estimate the intensity in equal area units using an appropriate statistical estimator, such as,[4] and to show how the intensity of SPPP varies. It may be constant ("uniform" or "homogeneous") or may vary depending on location ("spatial" or "inhomogeneous"). A Poisson point process, with a constant intensity λ, is also called Complete Spatial Randomness (CSR) in the literature.

Assuming the intensity of the PPP does not depend on space, the expectation of number of points in a given region B is given by:

$$\mathbb{E}(|X \cap B|) = \lambda \mathcal{A}(B)$$

because $|B|$ follows a Poisson distribution with intensity $\lambda\mathcal{A}(B)$. Let us suppose now the intensity depends on spatial location $\vec{x}$, it can be shown that:

$$\mathbb{E}(|X \cap B|) = \int_B \lambda(x) d\vec{x} \tag{2}$$

Figure 2 shows some simulated realizations of a Poisson Point Process with various values of intensity. Figure 3 shows some simulated realizations of an Spatial Poisson Point Process using various non constant intensity function depending on spatial position.

If the intensity is inhomogeneous, the intensity function can be estimated locally under the assumption of locally constancy. In the following subsection, we present such a method based on an image regular partitioning.

3.2 Quadrant Test

To identify a wild plantation, i.e a PPP, a χ^2 test denoted to test the hypothesis of distribution of two samples, is used on several subregions $B_1, B_2, ...B_m$ of equal area, called quadrants (or quadrats). Then, points belonging to each quadrant is counting: $n_j = n(X \cap B_j)\ for\ j = 1, ..., m$. Under the hypothesis of homogeneous Poisson Point Pattern (or CSR), the $\{n_j\}$ are independent and identically distributed.
For each subregions, we estimate the intensity function λ (under hypothesis of a CSR). To verify their spatial homogeneity, a set of samples pattern is generated using the estimated intensity function λ. Then, we check if the samples are significantly closed to the observation. To perform the comparison, the distance between two samples must be considered. Several distances can be used, such as:

1. the Pearson distance χ^2 goodness-of-fit test;[5]
2. the Clark Evans distance (CE):[6] it compares the mean distance between nearest neighbors to expected distance between nearest neighbors when the points are dispersed in a random pattern;
3. the index of dispersion (ID):[7–9] it is a quadrant based index computed on the frequency of events in each quadrant and the number of quadrants
4. the distance index of dispersion (I):[10] it is a distance based on point-to-event distances. $I = 2$ suggests a random pattern, $I < 2$ a uniform pattern and $I > 2$ a clustered pattern.

Quadrant based methods are very dependent of the space partition and it therefore does not show necessarily the spatial variation of the intensity function λ between two adjacent regions. Another method, described in the next Section allows us a visual interpretation of the spatial variation of the intensity function λ.

3.3 Kernel Estimation

A second approach is to estimate the intensity function λ using a kernel method such as.[11] This method computes a mass value for each point of the process in order to estimate λ, these values are then smoothed using a Gaussian convolution. This smoothing allows the visualization of λ disparities at various scale σ such as illustrated in Figure 4. Regions obtains by thresholding can provide good candidates for an image partition spatially homogeneous with respect to λ.

In order to analyze the interactions between neighbor points, a second order method should be used. This is described in the next Subsection.

3.4 Second order analysis

Ripley's K-function[2,3] is widely used as a second order descriptive statistic in two-dimensional point pattern analysis. This function analyzes all point-to-point distance, as opposing to a first order analysis like nearest distance. Ripley defines the K-function for a point process such as $\lambda K(r)$ is the expected number of other points of the process within a distance r. Under a CSR assumption, the expected number of points belonging to a circle centered on x with radius r is $\lambda \pi r^2$. The general form of the estimator of K is

$$\widehat{K}(r) = \frac{1}{\widehat{\lambda}^2 \mathcal{A}(S)} \sum_{i=1}^{n} \sum_{j \neq i}^{n} \mathbb{1}_{\{||x_i - x_j|| \leq r\}} e(x_i, x_j, r) \tag{3}$$

where $e(x_i, x_j, r)$ is an edge correction used for eliminating biais due to the edge effect. Several edge correction strategies are available in the literature, see[12] for a survey of edge effect corrections. S is the spatial domain and $\mathbb{1}$ is the indicator function defined as $\mathbb{1}_{\{||x_i - x_j|| \leq r\}} = \begin{cases} 1 \text{ if } ||x_i - x_j|| \leq r \\ 0 \text{ otherwise} \end{cases}$.

Under the assumption of CSR, we have $\widehat{K}(r) = \pi r^2$, otherwise values such as $\widehat{K}(r) < \pi r^2$ suggest clustering while values such as $\widehat{K}(r) > \pi r^2$ suggest a regular pattern. The interpretation of the K-function is often difficult. A commonly-used transformation of is the L-function[13] defined as:

$$L(r) = \sqrt{\frac{K(r)}{\pi}} \tag{4}$$

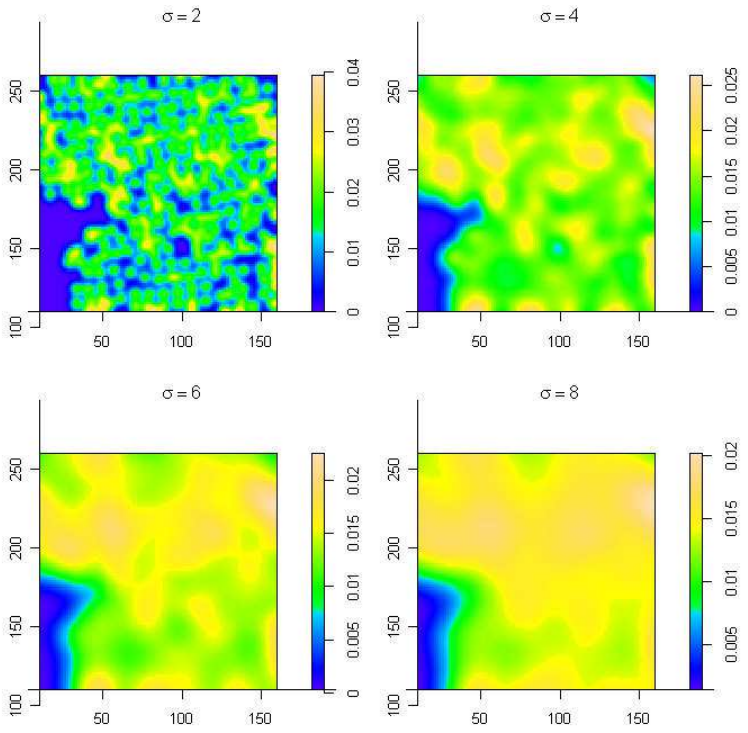


Figure 4. Various estimations of the intensity function $\lambda(u)$ using different values of the Gaussian kernel convolution.

Under the hypothesis of a CSR, we have $L_{pois}(r) = r$.

The Ripley's K-function is cumulative, so it becomes problematic to interpret the spatial interaction at larger distances. The pair correlation function $g(r)$[14, 15] is better for this purpose and for a stationary point process the $g(r)$ function is defined as:

$$g(r) = \frac{K'(r)}{2\pi r} \tag{5}$$

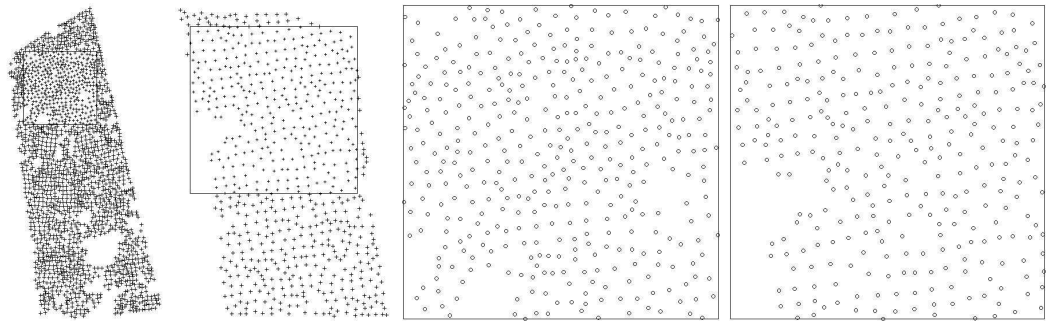
where $K'(r)$ is the derivative of $K(r)$. With a CSR hypothesis, we have $g(r) = 1$. In the opposite case, values such as $g(r) < 1$ suggest inhibition (or regularity) between points and values such as $g(r) > 1$ suggest clustering (or aggregation).

4. CHARACTERIZING COCONUT FIELDS

Our study focusses on two regions, denoted region #01 and #04 and presented in Figure 1. For each region, a subset of the point pattern is extracted within a square of 150m×150m (representing an area of 2.25 Ha). The minimum distance between each trees is $3.6m$ in Subregion #01 and $4m$ in Subregion #02. Then, applying various methods, we characterize the coconut plantations.
Figure 5 shows the full point pattern for regions #01 and #04, and for the corresponding subregions. Table 1 gives summary of the point pattern statistics like areas, number of trees and average intensities.

	Region #01	Subregion #01	Region #04	Subregion #04
Area (in Ha)	12.69	2.25	3.99	2.25
Number of points	1950	399	632	316
Average intensity	0.0154	0.0177	0.0158	0.014

Table 1. Some statistics from studied regions and subregions. The average intensity is the number of trees per square meter.

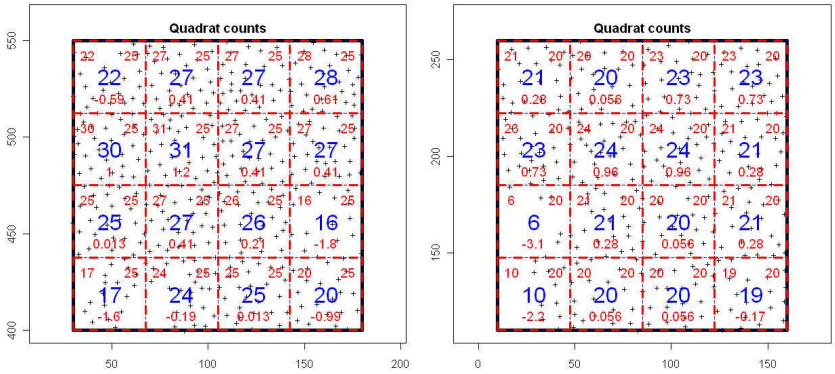


(a) Region #01 (b) Region #04 (c) Subregion #01 (d) Subregion #04

Figure 5. Studied regions and subregions.

The first step of the spatial analysis is to estimate the intensity of PPP. We use the methods described in Sections 3.2, 3.3 and 3.4.

First method is quadrant count. Subregions are divided into 4×4 equals areas and distances cited in Subsection 3.2 are computed. Figure 6 shows the partitioning in equals areas for each Subregions. Table 2 shows values of the various distances computed to determine if the point pattern reject the hypothesis of a CSR. Finally, this technique gives an indication on the intensity function $\lambda(\vec{x})$ in each areas of the partitioning, bit it is highly dependent of the number of quadrants used in the space partition.



(a) Subregion #01: $p - value = 0.76$ (b) Subregion #04: $p - value = 0.25$

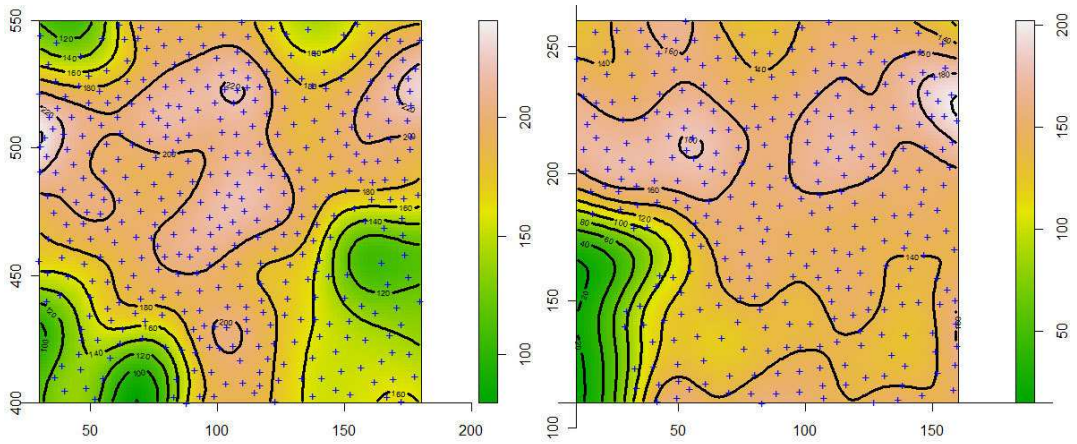
Figure 6. χ^2 test for the two subregions.

	Subregion #01			Subregion #04		
	CE	ID	I	CE	ID	I
Index Value	1.30	3.81	2.53	1.27	5.58	5.95
Test Statistics	8.76	3.81	2.53	7.26	5.58	25.20
Reject Ho	yes	no	yes	yes	yes	yes

Table 2. Determination of Indices of Dispersion: Clark Evans (CE), Index of Dispersion (ID), Distance Index of Dispersion (I).

We can see in Figure 7, the result of the kernel estimation with $\sigma = 12m$. It shows spatial variation of the intensity indicating a spatial PPP.

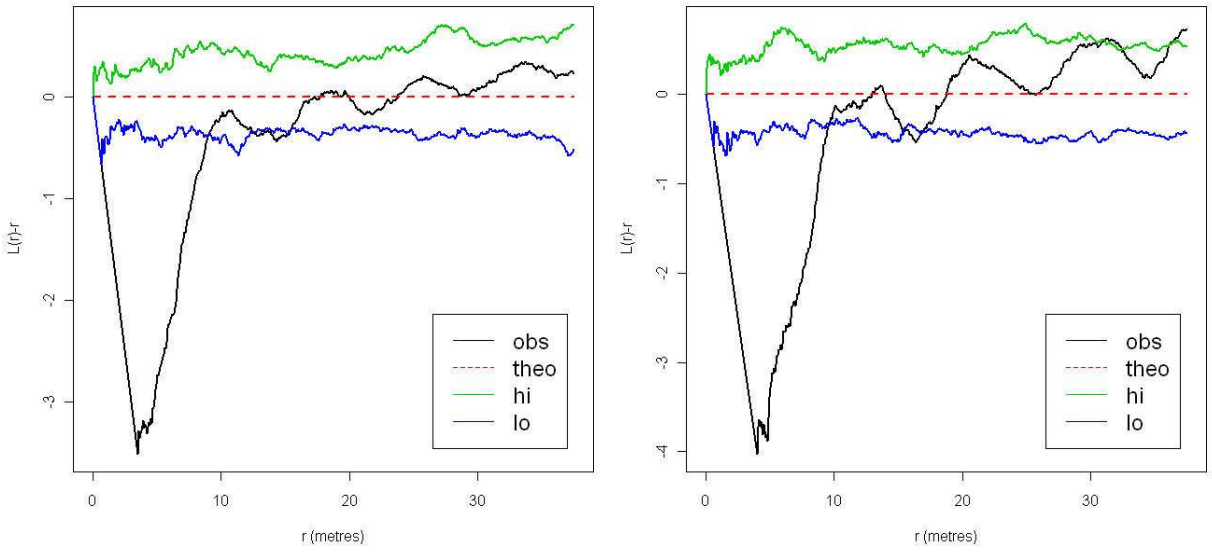
To extend our analysis, we compute the L-function as defined in equation (4) for the two Subregions. Results are shown in Figure 8 as $L(r) - r$. Under the hypothesis of CSR, $L(r) - r = 0$. If the point pattern exhibits aggregation, then $L(r) - r$ tends to be positive, and if it exhibits regularity, then $L(r) - r$ tends to be negative. Points in both Subregions #01 and #04 show a significant regularity for $r < 9.5m$ (the L-function computed with



(a) Kernel estimation on Subregion #01 (b) Kernel estimation on Subregion #04

Figure 7. Kernel estimation of the two subregions #01 and #04. Densities are plotted as number of trees per hectare.

the observation is under the lower confidence bound) with a minimum distance between each tree about $3.5m$ for the Subregion #01 and $4m$ for the Subregion #04. There is also a regularity observed for $r \in]11m, 15.5m]$ for Subregion #01. Finally, points of the Subregion #01 don't show deviation from CSR for $r \in [9m, 11m[$ and for $r > 15.5m$. Points in Subregion #04 exhibit a small clustering for $r \in [31m, 32m]$. For $r \in]9.5m, 31m]$, there is no deviation from CSR (the curve of the observation is between the lower and upper bounds of confidence) at larger scales (larger distances r).

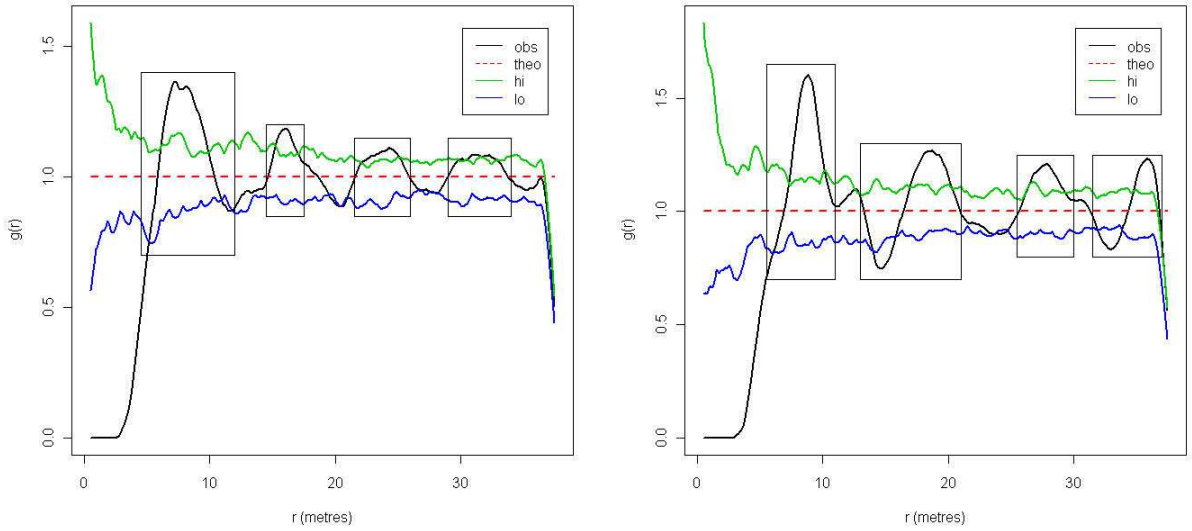


(a) L-function on Subregion #01. (b) L-function on Subregion #04.

Figure 8. L-function plotted as $L(r) - r$ with the upper (green line) and lower (blue blue) 95% confidence bounds, the theoretical curve (red dashed line) and the observation (black line).

The pair correlation function g for a point process defined as in equation (5) is also computed for the two subregions. Figure 9 displays such a function with lower (green line) and upper (blue line) confidence bounds plotted. Theoretical curves (red dashed line) and observed curves (black line) are also plotted. Under the hypothesis of CSR, $g(r) = 1$, values of $g(t) < 1$ suggest a regularity between points and values $g(r) > 1$ suggest clustering.

We observe that trees in Subregion #01 are clustered around distance values of $r = 6-10, 15-17, 23, 24-25m$, and those in the Subregion #04 are aggregated around distance values of $r = 7-10.5, 17-21, 26-29m$. Comparing Figures 8 and 9, we see that L-function and pair correlation function detect the same patterns but with different distance values.



(a) g function on Subregion #01. (b) g function on Subregion #04.

Figure 9. Pair correlation functions (g function) plotted with the upper (green line) and lower (blue line) 95% confidence bounds, the theoretical curve (red dashed line) and the observation (black line).

For this species of coconut trees (*Grands Locaux*), spacing between each tree should be $8.5m$, which implies a density of 140-150 trees per hectare.[16] According to these informations, we examine the density histogram and we define three classes of coconut plantations taking account the theoretical density and the standard deviation of the result of the kernel estimation (σ_{KD}), described as:

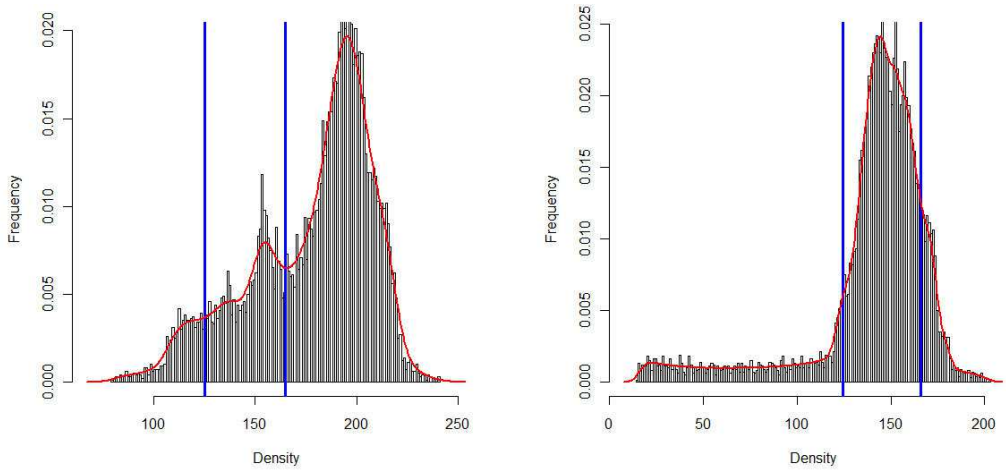
$$\begin{cases} values < 140 - \sigma_{KD} & \text{less wildness region: class 1} \\ 140 - \sigma_{KD} \leq values \leq 150 + \sigma_{KD} & \text{maintained fields: class 2} \\ values > 150 + \sigma_{KD} & \text{wild or natural fields: class 3} \end{cases} \tag{6}$$

Figure 10 shows the histogram of the estimated density for Subregions #01 and #04. The threshold values are plotted in the blue vertical line. Low and High threshold values for Subregion #01 are 125 trees/hectare and 165 trees/hectare. Those for Subregion #04 are 124 trees/hectare and 166 trees/hectare.

Figure 11 shows the result of the histogram threshold in three regions with over plotted trees.

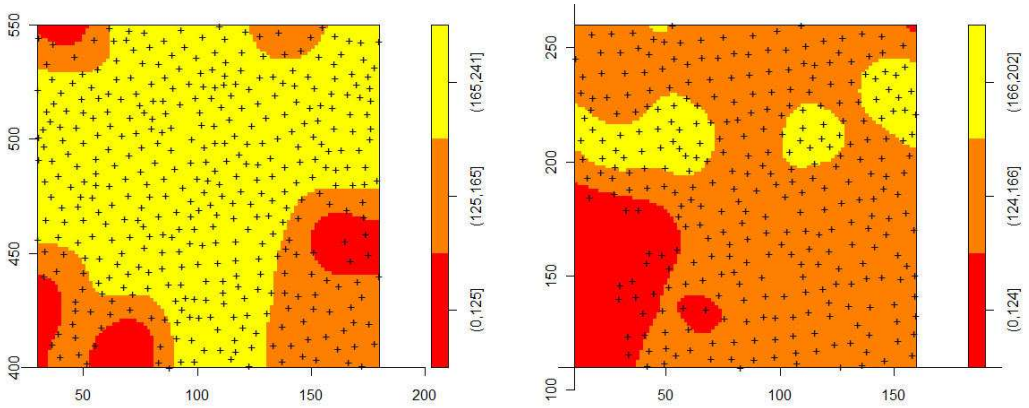
Table 3 lists statistics computed for the three classes, these classes being obtained by a histogram thresholding using values described in (6). Even if the area and the number of trees in class 3 (representing then the wild or natural plantation) of each subregion (Subregions #01 and #04) are different, the average intensity is the same. We observe the same conclusion for class 2 (classified as maintained plantations). Addressing class 1 in Subregion #01, there is only 7 trees detected inside. Even if the average intensity is closed to the intensity of class 1 in Subregion #04, we can not conclude for this region because the number of trees is not relevant. But it also indicates that maintained and wild plantations are mainly the two types of plantation in the Subregion #01.

In order to take a decision on the spatial properties of each segmented region, we use a Q-Qplot of residual. A Q-Qplot of residual is a diagnostic tool for checking some assumptions about the distribution. For example, a Q-Qplot of the (smoothed) residuals from a fitted model is a useful way to check the inter-point interaction.[17] Figure 12 shows the Q-Qplot of smoothed residuals for the three segmented regions on Subregions #04. For each plot, intervals with 95% of confidence are plotted in red dashed line and the model to fit in black dotted line.



(a) Kernel density histogram on Sample 01 (b) Kernel density histogram on Sample 04

Figure 10. The histogram of the kernel density estimation of Subregions #01 and #04. The vertical blue line are the threshold values.



(a) Thresholded kernel density of Subregion #01 (b) Thresholded kernel density of Subregion #04

Figure 11. Thresholded kernel density of Subregions #01 and #04.

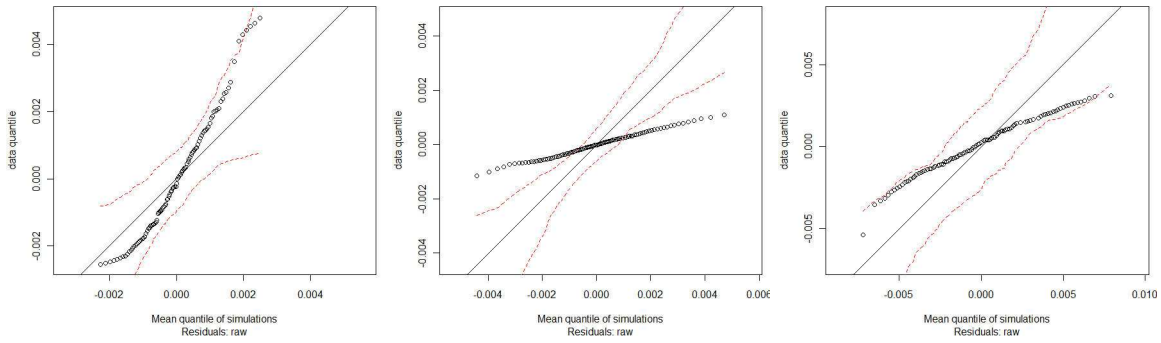
	Subregion #01			Subregion #04		
	Area 1	Area 2	Area 3	Area 1	Area 2	Area 3
number of trees	7	78	314	20	243	53
Average intensity (# trees/m^2)	0.00414	0.0154	0.02	0.00631	0.0148	0.0185
area (in m^2)	1692	5080.5	15727.5	3170.5	16465.5	2864.25

Table 3. Resume of each segmented regions for both Area 01 and Area 04.

Sub-figures 12(a) and 12(c) suggest that the uniform Poisson model is appropriate for points for classes 1 and 3 and we conclude that the coconuts' fields in the corresponding regions are wild. Class 3 has a higher degree of wildness than class 1 because its average intensity within class 3 is higher. Figure 12(b) indicates that the Poisson model is inappropriate for class 2 and can be characterized as a maintained plantation.

5. CONCLUSION

In this study, we classified a coconut trees plantation using a Spatial Poisson Point Pattern approach. Methods used allow us to characterize the spatial distribution of coconuts. The last part of this work consisted to segment the regions into three classes using a histogram analysis and a threshold on the result of kernel density estimation according to theoretical values of density of trees per hectare for a specific species of coconut trees. The analysis of these segmented regions gives information of the spatial distribution of trees.



(a) Q-Qplot for Area 1 (b) Q-Qplot for Area 2 (c) Q-Qplot for Area 3

Figure 12. Q-Qplot for the segmented regions of Subregion #04.

In a future work, the threshold values should be estimated using histogram approximation with a mixture of Gaussian.

REFERENCES

[1] Teina, R., Bériziat, D., Stoll, B., and Chabrier, S., "Toward a global Tuamotu archipelago coconut trees sensing using high resolution optical data," IGARSS08 (July 2008).

[2] Ripley, B. D., "Modelling spatial patterns," *Journal of the Royal Statistical Society, Series B* **39**, 172–212 (1977).

[3] Ripley, B. D., [*Spatial Statistics*], Wiley, New York, NY (1981).

[4] Diggle, P. J., [*Statistical Analysis of Spatial Point Patterns*], Academic Press (1983).

[5] Pearson, K., "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can reasonable be supposed to have arisen from random sampling," *Philosophical Magazine* **50**, 157–175 (1900).

[6] Clark, P. J. and Evans, F. C., "Distance to nearest neighbor as a measure of spatial relationships in populations," *Ecology* **35** (1954).

[7] Ludwig, J. A. and Reynolds, J. F., [*Statistical ecology: a primer on methods & computing*], John Wiley & Sons, Inc., New York, NY, USA (1988).

[8] Pielou, E., [*Mathematical Ecology*], John Wikey & Sons, New York (1977).

[9] Jayaraman, K., *A statistical manual for forestry research.* Forestry Statistics and Data Collection - DCA/MISC/01, Bangkok, Thaïland (Oct 2000).

[10] Johnson, R. B. and Zimmer, W. J., "A more powerful test for dispersion using distance measurements," *Ecology* **66**(5), 1669–1675 (1985).

[11] Diggle, P., "A kernel method for smoothing point process data," *Applied Statistics* **34**(2), 138–147 (1985).

[12] Baddeley, A. J., "Spatial sampling and censoring," in [*Stochastic Geometry: Likelihood and Computation, chapter 2*], 37–78, Chapman and Hall (1998).

[13] BESAG, J., "Contribution to the discussion of dr ripley's paper," *Journal of the Royal Statistical Society, Series B* **39**, 193–195 (1977).

[14] Stoyan, D. and Stoyan, H., "Random shapes and point fields, methods of geometrical statistics," (1994).

[15] Stoyan, D. and Stoyan, H., "Estimating pair correlation functions of planar cluster processes," *Biometrical Journal* **38**(3), 259–271 (1996).

[16] Chantry, G.; Cabannes, Y. W. V., [*Le cocotier: production et mise en oeuvre dans l'habitat*], Ministère de la Coopération, Paris (France) (1983). ISBN: 2-11-084847-2.

[17] Baddeley, A., Turner, R., Moller, J., and Hazelton, M., "Residual analysis for spatial point processes (with discussion)," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**, 617–666 (November 2005).

TOWARD A GLOBAL TUAMOTU ARCHIPELAGO COCONUT TREES SENSING USING HIGH RESOLUTION OPTICAL DATA

R. Teina[1], *D. Béréziat*[1,2]

[1]University Pierre and Marie Curie
Laboratoire d'Informatique de Paris 6
104 avenue du Président kennedy, F-75016 Paris
[2]Institut National de Recherche en Informatique
BP 105, F-78153 Le Chesnay Cedex

B. Stoll, S. Chabrier

Université de la Polynésie Française
Laboratoire GEPASUD
BP 6570, 98702 FAA'A
Tahiti - French Polynesia

ABSTRACT

This study is part of a regeneration program of the coconut grove of French Polynesia where most coconut palm trees of the Tuamotu archipelago were planted in the 1980's following the various hurricanes that had struck islands. The French Polynesia government acquired one-meter pansharpened RGB Ikonos images over the Tuamotu archipelago. To exploit these data, a pilot study is conducted on the island of Tikehau, well-known from the specialists and easily accessible from Tahiti. A Maximum Likelihood (ML) classification is performed to segment the high vegetation in images. Thus, a Support Vector Machines (SVM) classification allows the high vegetation to be classified in different patterns. And finally, a robust segmentation process based on markers controlled watershed segmentation is proposed to extract tree crows. Through the ground mission, the trees detection accuracy is estimated which is then used to compute the number of trees the closest to the reality by applying a weighted factor to the number of trees located in each class.

Index Terms— SVM, Maximum Likelihood, classification, segmentation, watershed, texture.

1. INTRODUCTION

French Polynesia government wants to improve the coconut tree field exploitation in order to develop the extraction of the Coprah oil as an alternative fuel and also the use of senile trees wood. The Coprah oil exploitation already constitutes one of the principal financial resources of the inhabited atolls. Remote sensing technics minimize the cost of these studies by automating this task and avoiding ground missions.

The aim of this work is to characterize the coconut field type typology, *i.e.* the spatial distribution of coconuts. This goal is reached by first segmenting the coconut trees and then by classifying them according to their spatial distribution. Several steps are required to perform the automatic enumeration of coconut trees: a coconut field segmentation keeping the area of interest, a coconut field classification among a set of classes describing the different type of coconut fields encountered in Tuamotu using texture features and the SVM classifier, and finally, a robust segmentation process of coconut trees crown. An enumeration is now possible and an identification / segmentation of these trees according to some objective criteria such as the size of canopy, the average color, local density of the coconut trees field. A ground truth validation is performed in order to estimate the detection rate and error in each coconut trees class type leading to a precise extrapolation of the global number of trees. The convergence between results obtained with the proposed method and ground truth missions highlights the robustness of the method and authorizes a large-scale production on all islands.

2. DATA AND SITE STUDY

IKONOS optical data is widely available through the whole Tuamotu archipelago and its high spatial resolution (about one-meter resolution at ground level) is sufficient to carry out our objective. The study focuses on the atoll of Tikehau that is well-known from the specialists and easily accessible from Tahiti as a validation study area before extending the method to the rest of Tuamotu's atolls. Tikehau data set was acquired by IKONOS2 on July and August 2003 and is already orthorectified and registered in the WGS84 projection. The atoll of Tikehau is constituted of several islands called *motu*. As the full mosaic of the atoll of Tikehau has a resolution of 28517 by 28617 pixels, the original image is cut out into sub-images, each one locating a *motu*.

3. TREE FIELDS CLASSIFICATION METHODOLOGY

The coconut trees crown segmentation process must be applied in coconut fields areas to avoid false alarms and reduce the number of pixels to classify. In images, several structures are distinguished such as the sea, the sand, the coral and

some dwellings as well as the vegetation (coconut trees and other atoll vegetation types). In a first step, it is necessary to generate high vegetation masks before applying the segmentation process. Due to the lack of the near infra-red (NIR) band (not available in our database), it is not possible to compute the well known normalized difference vegetation index (NDVI) which is relevant for characterize the vegetation. An alternative solution has been tested and chosen: a Bayesian classification using a ML algorithm. However, this implies a manual selection of the training sets for each different available structures in the image. Then, a texture analysis [1, 2] is performed on the vegetation class to separate the high vegetation (which has been proved to be exclusively coconuts in Tuamotu) from the low vegetation. Once the segmentation is completed, areas are classified in three types of planting: natural (non spatial organization), artificial (trees are positioned on a grid spaced by 8 meters) and mixed fields (artificial fields where activities were dropped). The tree crown segmentation process depends on a pixel's neighborhood to compute the minimal network of the darkest pixels. The planting type classification is needed to adjust the size of the research window. For this case, a supervised SVM classification is used to segment these fields based on some training sets representing each kind of these fields that were selected in order to extract features like texture information [2, 3, 4, 5].

3.1. SVM Principle

The SVM classifier was introduced in 1995 by [6]. This Subsection briefly describes this classifier, details can be found in [7]. The Figure 1 illustrates the principle of the optimal hyperplan and the optimal margin used in the SVM classifier.

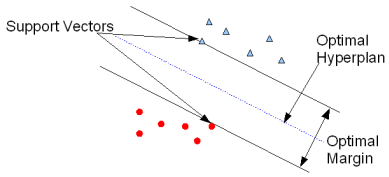


Fig. 1. SVM Classifier principle

Let's consider the case of a two-classe classification problem. A training set data is constituted of N samples described by data X_i and the labels Y_i take -1 or $+1$ as values. For the linear separability data, the SVM classifier is defined as the function

$$f_i(x) = sign(\langle \omega, X_i \rangle + b) \quad (1)$$

which maximizes the margin between the optimal hyperplan and the support vectors. The problem can be solved by using the Lagrange multipliers

$$f(x) = sign\left(\sum_{i=1}^{N} y_i.\alpha_i \langle x, x_i \rangle + b\right) \quad (2)$$

where α_i are Lagrange multipliers.

In the case of nonlinear separable data, one possible method to solve the problem is to use a kernel. This kernel is a function which project the initial data into a higher dimension space feature, $\Phi : \mathbb{R}^n \rightarrow \mathbb{F}$, in a such way that data are now considered as linearly separable. In equation (2), the dot product $\langle x, x_i \rangle$ is replaced by the dot product associated to the space feature $\mathbb{F}$ defined as:

$$K(x, x_i) = \langle \Phi(x), \Phi(x_i) \rangle \quad (3)$$

and then the function to classify data becomes

$$f(x) = sign\left(\sum_{i=1}^{N} y_i.\alpha_i.K(x, x_i) + b\right) \quad (4)$$

In this study we use the Radial Basis Kernel (RBF) defined with $K(x, x_i) = e^{-\frac{|x-x_i|^2}{2\sigma^2}}$. This kernel gives the best results in our case compared to other kernels we have tested.

3.2. Application to Ikonos images

We use a modified ground truth made by Jacq [8] on the main *Motu* to describe our classes. For each class, an equalized random samples of 1100 pixels is taken for the training set as described in [9]. The definition of training set classes is listed in Figure 2. The SVM classification was performed using [10, 11] available in ENVI Software on a stack composed by the RGB components and the normalized Haralick's texture descriptors (with a total of 11 bands). The classification result has been improved using a majority analysis with a 45×45 window size. The Figure 2 shows the final fields type classification on the main *motu* of Tikehau.

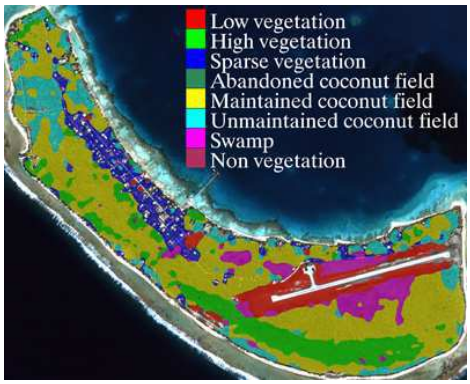


Fig. 2. SVM classification result on the main *motu* and classes color.

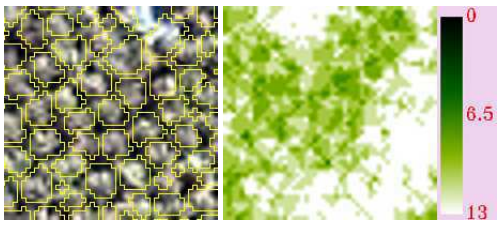
4. TREE CROWN DELINEATION METHODOLOGY

The crown delineation method is based on [12] which uses a watershed segmentation to extract tree crowns. However, several modifications has been bring to take into account specificity of image and trees: after an anisotropic filtering [13]

of high vegetation areas on the first PCA component [14] of the RGB image, local maximums are detected and used as markers for the watershed; these maximums are located closed to the crowns center [15]. The assumption that there are the darkest pixels between each crown is used to compute a boundary crown network [16, 17]. This minimal network of dark pixels is computed as the local minima of a searching window which the size depends on the planting type. The minimal network of dark pixels and markers are used as new minimum for the watershed segmentation process and to compute a topographic surface using the L_2 distance map. To avoid too small regions, we discriminate them according to their area using a coconut tree crown approximation by a circle within a radius R. Once the crowns delineation is made, the center of mass, weighted by the values of the pixels in each RGB channel, is computed to estimate the center of canopies. For details, the reader is referred to [18].

5. RESULTS, VALIDATION AND EXTRAPOLATION

In order to validate the whole method, some sample areas characterizing various fields type are chosen. For each of these sample areas, the segmentation process is executed and all tree crowns are extracted. The validation is made using a photo ground truth and a ground mission. The Figure 5 shows a result of the crown segmentation on a small region and the corresponding local density map.



(a) Tree crown delineation (b) Density Map

5.1. Validation

A first validation is performed on images (acquired in 2003) by human interpretation (manual localization of the coconut trees canopy) in order to estimate the detection error for each kind of fields type. This validation provides a mean detection error of less than 10% (see table 1) which proves that our method locates a majority of the coconut trees visible in images.

A ground mission was planed in 2006 by Jacq [8] which has counted coconut trees within 12m radius circular plots, each plot being located by its GPS coordinates. Moreover, the tree height had been recorded. This mission had covered the entire atoll of Tikehau. To compare our results with the ground mission results, the tree delineation process is used to compute a local density map with the same radius circle whose centers are given by GPS coordinates. Two similar local density maps are build from the ground mission data considering trees having more than 4-meter height and trees having more than 5-meter height, these heights representing trees likely visible on the remote sensed images. The Table 2 shows that our method detects between 66% to 89% (for H $>$ 5m) of the visible trees according in which class the trees are. In maintained fields, where trees density is the highest (distance between each tree is about 8m), the accuracy is closed to 83%. The zeros values for classes 3 and 8 indicate that no measurements have been made in this regions during the different ground missions.

Table 1. Method accuracy using a photo interpretation ground truth

Id Area	# Method	Photo	accuracy
1	908	818	9,91%
2	986	912	7,51%
3	495	450	9,09%
4	346	356	-2,89%
5	436	399	8,49%
6	1269	1198	5,59%
Total	4440	4133	6.81%

Differences between validations come from the fact that the photo interpretation is done on the image taken in 2003. The ground truth missions was planed in 2006, 3 years later. During this period of 3 years, some trees not visible on the image may have arrived to the top of the canopy, also other trees may have been able to grow (especially in unmaintained fields) and some may have died.

5.2. Trees Extrapolation

Once the method's accuracy is estimated for each class, we can estimate a number of coconut trees in each class and on all over the atoll of Tikehau. We believe it is wise to take as precision for each class an average accuracy from all trees, those over 4-meter height and those over 5-meter height (see the last column in Table 2). Excepted for classes 3 and 8 whose the accuracy can not be measured, we only keep the same number of detected trees for those classes. The estimated number is computed by applying a weighting factor to the number of trees located in each class and summing them to obtain a global estimation.

6. CONCLUSION

Despite of the lack of the NIR band, combining the use of ML and SVM classification on Haralick's texture descriptors provides a good classification of various coconut field types. The segmentation process based on a markers controlled watershed algorithm gives good results since that the photo interpretation shows less than 10% of false detections. Our tree crown segmentation method is robust and allows a production on all atolls. The ground truth surveys in each class provides an accuracy of good detection for each type of planting. Ap-

Table 2. Ground truth accuracy on the entire atoll

Class Id	number of trees				Accuracy			
	all	H > 4m	H > 5m	Method	% Diff 4m	% Diff 5m	% Diff All	% Mean
1	73	50	48	51	102,00%	106,25%	69,86%	92,70%
2	228	206	185	132	64,08%	71,35%	57,89%	64,44%
3	0	0	0	0	0,00%	0,00%	0,00%	0,00%
4	687	584	540	399	68,32%	73,89%	58,08%	66,76%
5	174	133	123	102	76,69%	82,93%	58,62%	72,75%
6	8	6	6	4	66,67%	66,67%	50,00%	61,11%
7	22	21	16	11	52,38%	68,75%	50,00%	57,04%
8	0	f0	0	0	0,00%	0,00%	0,00%	0,00%

Table 3. Final coconut trees estimation all over the island of Tikehau

	Class Id								
	1	2	3	4	5	6	7	8	
Mean accuracy	92,70%	64,44%	0,00%	66,76%	72,75%	61,11%	57,04%	0,00%	Total
Detected	13771	21735	650	87824	23439	1334	973	904	150630
Extrapolated	14855	33728	650	131546	32220	2183	1706	904	217792

plying different weighting factor according to a class, we obtain an estimated number of coconut trees the closest to the reality. A drawback of the classification method is its high computational cost as the classification is directly performed on pixel values leading to a big configuration space. To fix this problem, we propose to perform the classification of coconut plantation type directly on the crown center coordinate which is a small configuration space. A relevant representation of these coordinates using wavelet or Poisson process is probably required to obtain good classification results.

Acknowledgement

The authors would grateful to LIP6 for the available office, GePaSud for the ground truth mission, Fred Jacq for its data and would like to thank French Polynesia Gourvenment and its Urbanism Department for providing Ikonos data on the Tuamotu's islands.

7. REFERENCES

[1] H. Anys, H. Bannari, D.C. He, and D. Morin, "Cartographie des zones urbaines à l'aide des images MEIS-II," *International Journal of Remote Sensing*, vol. 19, no. 5, pp. 883–894, 1998.

[2] A. Puissant, J. Hirsch, and C. Weber, "The utility of texture analysis to improve per-pixel classification for high to very high spatial resolution imagery," *International Journal of Remote Sensing*, vol. 26, no. 4, pp. 733–745, February 2005.

[3] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, pp. 610–621, November 1973.

[4] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, pp. 786–804, 1979.

[5] H. Anys, H. Bannari, D.C. He, and D. Morin, "Texture analysis for the mapping of urban areas using airborne MEIS-II images," Strasbourg, France, 1994, number 3, pp. 231–245.

[6] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[7] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[8] F. Jacq, "Évaluation quantitative et qualitative des peuplements de cocotiers sur tikehau," Tech. Rep., Service du Développement Rural - Département FOGER, Avril 2006.

[9] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis: An Introduction*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1999.

[10] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[11] T.-F. Wu, C.-J. Lin, and R. C. Weng, "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research*, vol. 5, pp. 975–1005, 2004.

[12] L. Wang, P. Gong, and G.S. Biging, "Individual tree-crown delineation and treetop detection in high-spatial-resolution aerial imagery," *Photogrammetric Engineering and Remote Sensing*, vol. 70, no. 3, pp. 351–357, March 2004.

[13] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, 1990.

[14] C. Ricotta and C. Avena, "The influence of principal component analysis on the spatial structure of a multispectral dataset," *International Journal of Remote Sensing*, vol. 20, no. 17, pp. 3367–3376, 1999.

[15] R.J. Pollock, *The automatic Recognition of Individual trees in Aerial Images of Forests Based on a Synthetic Tree Crown Image Model*, Ph.D. thesis, University of British Colombia, Vancouver, Canada, 1996.

[16] F.A. Gougeon, "Automatic individual tree crown delineation using a valley-following algorithm and a rule-based system," in *Proceeding of Int. Forum on Automated Interpretation of High Spatial Resolution Digital Imagery for Forestry*, D.A Hill and D.G Leckie, Eds., Victoria, British Columbia, Canada, February 1998, pp. 11–23.

[17] D.S. Culvenor, "TIDA: an algorithm for the delineation of tree crowns in high spatial resolution remotely sensed imagery," *Comput. Geosci.*, vol. 28, no. 1, pp. 33–44, 2002.

[18] R.Teina, D. Béréziat, B. Stoll, and S. Chabrier, "Étude pilote de la cocoteraie de tikehau sur des images Ikonos," Caen, 2007, MajecSTIC 07, vol. 2, pp. 265–269.

A comparative study of several supervised classifiers for coconut palm trees fields' type mapping on 80cm RGB pansharpened Ikonos images

Teina R.[a] and Béréziat D.[a] and Stoll B.[b] and Chabrier S.[b]

[a]Université Pierre et Marie Curie, LIP6, 104 avenue du Président Kennedy, 75016 Paris, France;
[b]Université de la Polynésie Française, Laboratoire GePaSud, 98702 Faa'a, Tahiti - French Polynesia;

ABSTRACT

The purpose of this study is to classify the types of coconut plantation. To do this, we compare several classifiers such as Maximum Likelihood, Minimum Distance, Parallelepiped, Mahalanobis and Support Vector Machines (SVM). The contribution of textural informations and spectral informations can increase the separability of different classes and then increase the performance of classification algorithms. To compare these algorithms, the search for the optimal size of the window to the calculation of texture information is necessary, as well as optimization of parameters for SVM. Following this study, we find that SVM give very satisfactory results for coconut field type mapping.

Keywords: SVM, Maximum Likelihood, Mahalanobis, Parallelepiped, Minimum Distance, Texture, Principal Component Analysis

1. INTRODUCTION

French Polynesia government wants to improve the coconut tree field exploitation in order to develop the extraction of the Coprah oil as an alternative fuel and also the use of senile trees wood. The French Polynesia Government acquired in 2003 some RGB pansharpened Ikonos images 80 cm pixel resolution on ground for almost the 80 islands in the Tuamotu's archipelago. The goal of our work is to classify the coconut palm trees fields' using several supervised classification algorithms and to measure their accuracies. The several compared classifiers are Maximum Likelihood, Minimum Distance, Mahalanobis Distance, Parallelepiped and Support Vector Machines (SVM). First, the classifications are done using only the spectral informations but this is not enough to achieve our goal. Then, the Haralick's texture descriptors are computed using the first component of the Principal Component Analysis applied to the spectral bands. These texture descriptors are computed using a window size estimated by analyzing different field types. Spectral and textural informations are used for classifications. The classifiers' accuracies is done by analysis of the confusion matrix using a ground truth ROIs. According to the performance analysis, the use of SVM provides a good land use mapping of coconut field types.

2. DATA AND STUDY SITE

The study focuses on the atoll of Vahitahi, which is located in the north-east of the French Polynesia Territory, especially on the area located at the north-east where different types of plantation are available. The remote sensed image was acquired in September 2005 at 19:46 GMT by the satellite Ikonos and the data are projected in the WGS-84 coordinates system. The complete mosaic (10673 $\times$ 4120 pixels) is centered on geographic coordinates $138°51'52.96''W$ and $18°45'35.19''S$. The RGB images are pansharpened with a ground resolution of the pixel of 80 cm. The figure 1 illustrates a global view of the atoll of Vahitahi and a zoom of the study area in the north-east.

Further author information: (Send correspondence to Teina R.)
Teina R.: E-mail: Raimana.Teina@lip6.fr, Telephone: +33 (0) 1 44 27 88 16
Béréziat D.: E-mail: Dominique.Bereziat@lip6.fr

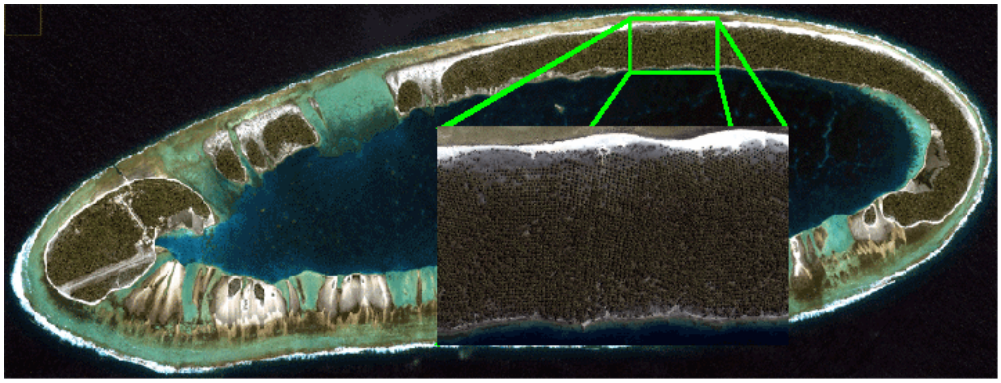


Figure 1. The atoll of Vahitahi and a zoom of the study site on the north-east.

3. CLASSIFIERS REVIEW

3.1 Maximum Likekihood Classification

3.1.1 Bayes' Classification

Let the spectral classes be represented by $\omega_i,\ i = 1, ..., M$, where M is the number of classes. The conditional probabilities $p(\omega_i|x)$ gives the likelihood that the correct class is ω_i for a pixel at location x. The classification is performed according to

$$x \in \omega_i, \text{if } p(\omega_i|x) > p(\omega_j|x) \text{ for all } j \neq i \tag{1}$$

The pixel at position x belongs to class ω_i if $p(\omega_i|x)$ is the largest.

3.1.2 Maximum Likelihood decision rule

The $p(\omega_i|x)$ in equation 1 are unknown but the $p(x|\omega_i)$ can be estimated from the training data. Using the Bayes' theorem (Freund[1]), we have:

$$p(\omega_i|x) = p(x|\omega_i)p(\omega_i)/p(x) \tag{2}$$

Using equation 2, the Maximum Likelihood (ML) classification's rule defined in equation 1 becomes

$$x \in \omega_i, \text{if } p(x|\omega_i)p(\omega_i) > p(x|\omega_j)p(\omega_j) \text{ for all } j \neq i \tag{3}$$

and the disciminant function $g_i(x)$ is defined as

$$g_i(x) = \ln p(x|\omega_i) \ln p(\omega_i) \tag{4}$$

In the case of a multivariate normal class models, the discriminant function $g_i(x)$ becomes

$$g_i(x) = -\ln |\Sigma_i| - (x - m_i)^t \Sigma_i^{-1} (x - m_i) \tag{5}$$

where m_i and Σ_i are the mean vector and covariance matrix of the data in class ω_i.

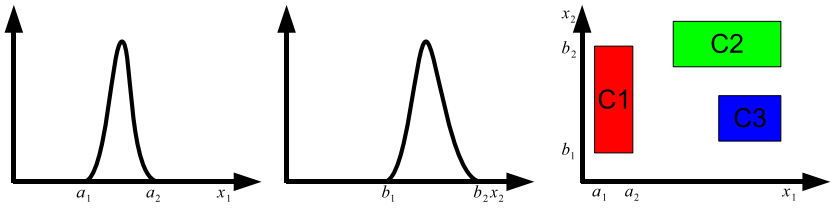
3.2 Minimum Distance Classification

Suppose $m_i, i = 1, ..., M$ are the means of the M classes determined from the training data, and x is the location of the pixel to be classified. The set of square Euclidean distances of the unknown pixel to each of the class means is computed as $d(x, m_i)^2 = (x - m_i)^T (x - m_i)$. The Minimum Distance (MinDist) classification is performed on the base of the rule described below:

$$x \in \omega_i, \text{if } d(x - m_i)^2 < d(x - m_j)^2 \text{ for all } j \neq i \tag{6}$$

3.3 Parallelepiped Classification

The parallelepiped (Para) classifier is trained by inspecting histograms of the individual spectral component of the available training data. The lower and the upper significant bounds on the histogram are identified and used to describe the brightness value range for each band for that class. The range in all bands describes a multidimensional box or parallelepiped. During the classification process, if pixels are found to lie in such a parallelepiped they are labelled as belonging to that class.



(a) Histograms with upper and lower bounds.

(b) Examples of 2D parallelepipeds.

Figure 2. Principle of the Parallelepiped classification.

3.4 Mahalanobis Classification

Consider the discriminant function defined in equation 5 and let reverse the sign of this function, we have

$$d(x, m_i)^2 = \ln|\Sigma_i| + (x - m_i)^t \Sigma_i^{-1} (x - m_i) \tag{7}$$

Now let consider the case where all class covariances are equal, i.e $\Sigma_i = \Sigma$ for all i, The term $\ln|\Sigma|$ can be ignored. The Mahalanobis (Mahal) distance classifier is defined using the following distance measure:

$$d(x, m_i)^2 = (x - m_i)^t \Sigma^{-1} (x - m_i) \tag{8}$$

3.5 Support Vector Machine (SVM)

The SVM classifier was introduced in 1995 by Vapnik.[2] This Subsection briefly describes this classifier, details can be found in the works of Burges[3] and Wu et al.[4] Figure 3 illustrates the principle of the optimal hyperplan and the optimal margin used in the SVM classifier.

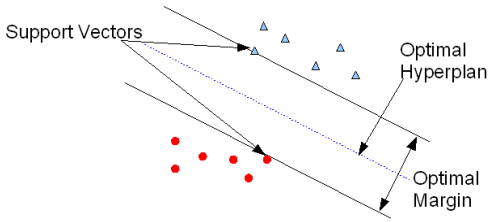


Figure 3. SVM Classifier principle.

Let's consider the case of a two-classe classification problem. A training set data is constituted of N samples described by data X_i and the labels Y_i take -1 or $+1$ as values. For the linear separability data, the SVM classifier is defined as the function

$$f_i(x) = sign(\langle \omega, X_i \rangle + b) \tag{9}$$

which maximizes the margin between the optimal hyperplan and the support vectors. The problem can be solved by using the Lagrange multipliers

$$f(x) = sign\left(\sum_{i=1}^{N} y_i.\alpha_i \langle x, x_i \rangle + b\right) \tag{10}$$

where α_i are Lagrange multipliers.

In the case of nonlinear separable data, one possible method to solve the problem is to use a kernel. This kernel is a function which project the initial data into a higher dimension space feature, $\Phi : \mathbb{R}^n \rightarrow \mathbb{F}$, in a such way that data are now considered as linearly separable. In equation 10, the dot product $\langle x, x_i \rangle$ is replaced by the dot product associated to the space feature $\mathbb{F}$ defined as:

$$K(x, x_i) = \langle \Phi(x), \Phi(x_i) \rangle \tag{11}$$

and then the function to classify data becomes

$$f(x) = sign\left(\sum_{i=1}^{N} y_i.\alpha_i.K(x, x_i) + b\right) \tag{12}$$

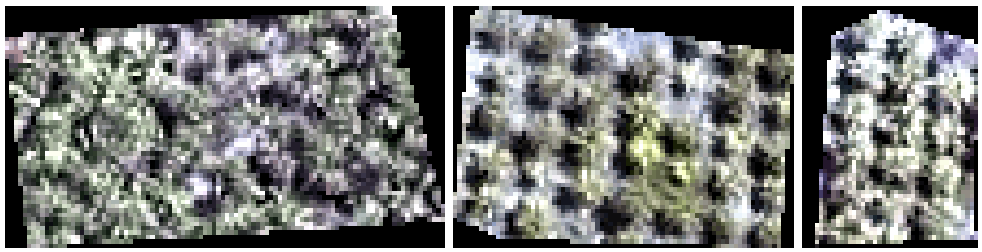
In this study we use the Radial Basis Kernel (RBF) defined with $K(x, x_i) = e^{-\frac{|x-x_i|^2}{2\sigma^2}}$. This kernel gives the best results in our case compared to other kernels we have tested.

4. METHODOLOGY

4.1 Training sets

The goal of the study is to analyze the accuracy of the supervised classifiers for the characterization of different planting types. We first segment the vegetation part where we define regions of interest (ROI) corresponding to each class of fields. As recommended by Swain and Davis,[5] $100N$ (with N the number of bands) training pixels per class are needed to avoid the covariance matrix being singular. In our study, training sets are constructed by randomly generating 1500 pixels in each ROI. Figure 4 shows three class of field types listed below:

- 4(a): natural field where there is not any spacial organization between trees (Natural),
- 4(b): artificial field with low density (Artificial LD),
- 4(c): artificial field with medium density (Artificial MD).



(a) Natural (b) Artificial LD (c) Artificial MD

Figure 4. Training set for each class of plantation.

To predict the result of the final classification, we computed the classes separability with the Jeffries-Matusita (JM) distance. This measure is the average distance between two classes density function (Wacker[6]) and for normally distributed classes it is defined as

$$J_{ij} = 2(1 - e^{-B}) \tag{13}$$

where B is referred to the Bhattacharyya distance (Kailath[7]) and is defined as

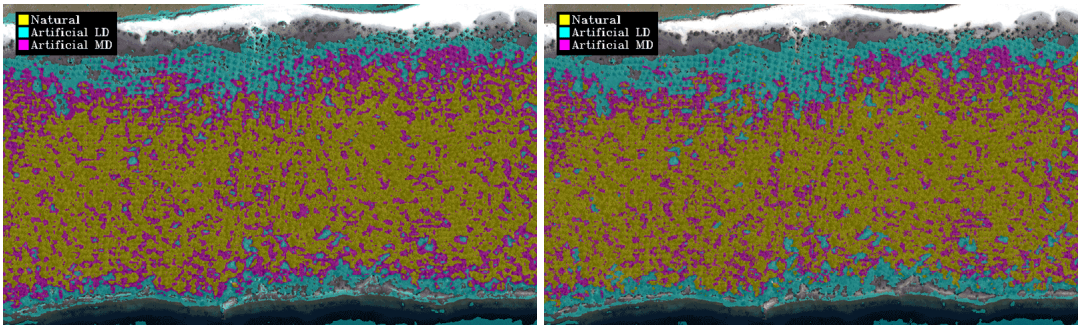
$$B = \frac{1}{8}(m_i - m_j)^t \left(\frac{\Sigma_i + \Sigma_j}{2}\right)^{-1} (m_i - m_j) + \frac{1}{2} \ln\left(\frac{|(\Sigma_i + \Sigma_j)/2|}{\sqrt{\Sigma_i \Sigma_j}}\right) \tag{14}$$

Table 1. Jeffrey-Matusita distance for each class.

	Natural	**Artificial LD**	**Artificial MD**
Natural	0		
Artificial LD	1.44641	0	
Artificial MD	0.74986	0.40689	0

As shown in Tab. 1, the use of only spectral informations is not enough to achieve a good classification. We can see that the best pair separation doesn't exceed 1.446 and at least Artificial LD and Artificial MD only have a pair separation equal to 0.407.

The spectral information is not sufficient to characterize the fields types. Indeed the figure 5 below, where classifications have been made only on spectral informations with ML and SVM algorithms, shows unsatisfactory results.



(a) ML classification (b) SVM classification

Figure 5. ML and SVM classification's results using only the spectral informations.

To achieve our task, we will use some textural informations defined by Haralick.

4.2 Textural informations

A first step is to do a Principal Component Analysis (PCA) on the RGB image. Then, all the texture descriptors are computed using the first component of the PCA transformation because following Ricotta et al.[8] it contains informations of the three bands (Red,Green and Blue) and has the higher variance

The eight texture descriptor described by Haralick[9–12](GLCM) and used in this work are: mean, variance, homogeneity, contrast, dissimilarity, entropy, second moment and correlation. As the artificial plantation have no preferred direction, each texture descriptor is computed in the four directions ($0°, 45°, 90°$ and $135°$) than a mean descriptor is calculated by averaging the four directions. The textural informations and the spectral informations are stacked together and normalized.

Figure 6 shows the evolution of the global mean of JM distance versus the size of the window used for the calculation of the texture descriptors. As shown in this figure, the global mean class separability is greater than 1.9 from a window size greater or equal to 29 pixels. This indicates that we could obtain a better accuracy of the classification from a window size of 29 pixels.

5. APPLICATION ON IKONOS IMAGES

In this section, we apply different classifiers reviewed in Sect. 3 using even spectral and textural informations.

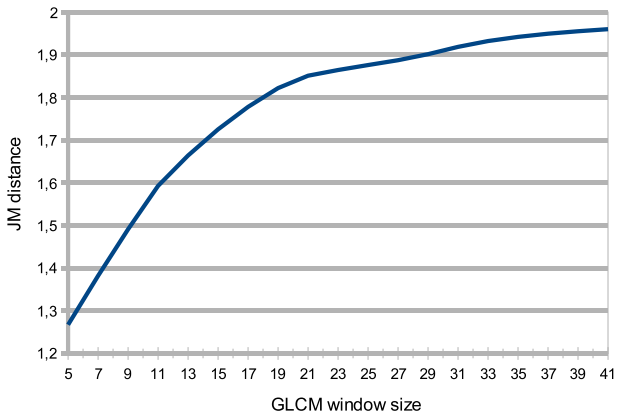


Figure 6. JM distance versus the window size used for the calculation of the texture description.

5.1 Comparison of different classifiers

For all those methods, we apply the classification with different window size (from 5×5 to 41×41 with a step of 2) used for the calculation of the textural informations. The accuracy is computed using the coherence matrix computed on 3000 randomly generated pixel's locations in ground truth ROI for each class.

For each method's classification result, the accuracy is compared by analyzing the mean accuracy, i.e the average of the diagonal values of the coherence matrix. Tab. 2 shows sample calculation of the mean accuracy (in %).

Table 2. Method to compute the mean accuracy using the confusion matrix.

	Natural	Artificial LD	Artificial MD	
Natural	**98,63**	0	14,33	
Artificial LD	0	**99,89**	16,53	
Artificial MD	1,37	0,11	**69,13**	Mean accuracy
				89,22

Figure 7 shows a plot of the classifiers' accuracies versus the GLCM's window size. As we can see, SVM's and Maximum Likelihood's accuracies are clearly better than the other classifiers. From a GLCM's window size equal 5 to 29, SVM and ML have similar accuracies. But while the GLCM's window size increase, SVM gives some better classification's results than the ML.

Considering the three other classifiers, we observe that Mahalanobis classification's accuracies tend to increase and reach its maximum value for a GLCM's window size of 37 pixels with a mean accuracy value of 74.54%. It shows the influence of textural informations for the classification's performance with a maximum difference between values equal to 24%.
As regards the Parallelepiped algorithm, the curve tends to grow until the GLCM's window size equal 37 pixels where the maximum accuracy is reached with a value of 63.2%.
As against, the Minimum Distance algorithm is one that gives the worst performance. Indeed, we see that the curve has a low slope, although it is positive. And textural informations do not improve more than 7% the accuracies of this method.

Looking at the results, we can distinguish three groups:

- Best performance: SVM and ML,
- Medium performance: Mahalanobis,
- Poor performance: Minimum Distance and Parallelepiped.

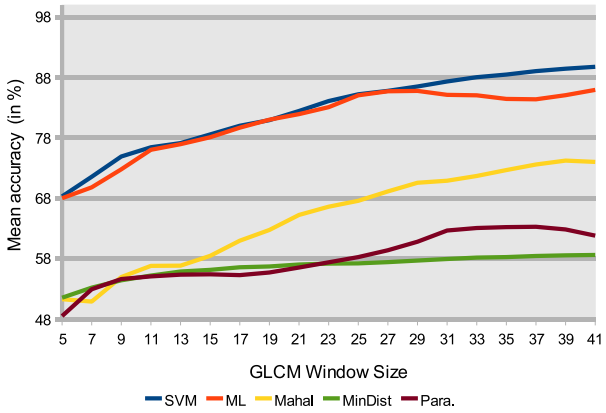


Figure 7. Classifiers' accuracy.

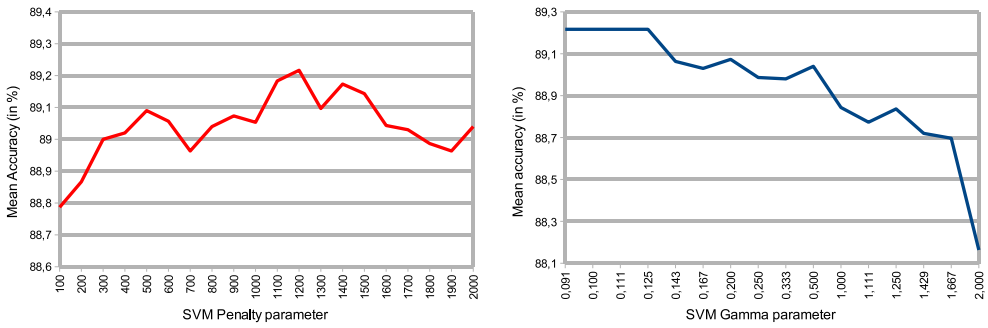
The SVM classification is done using the LIBSVM[13] and the default values, i.e using $\gamma = \frac{1}{N}$ with N the number of bands (3 spectral bands and 8 texture descriptors), and the penalty set to 100. Now we want to see if the SVM's parameter (Penalty and Gamma) can increase the mean accuracy of the classification in the next section.

5.2 SVM parameters optimization

To highlight the contribution of the SVM's parameters to the accuracy of the classification, we fix a GLCM's window size to 37. The data to classify are still the textural informations and the spectral informations stacked together.

First, we increase the penalty (P) parameter from 100 to 2000 with a step of 100. In Fig. 8(a), the mean accuracy is plotted for each penalty parameter. Variations of the parameter P does not show any significant increase in performance. Indeed, accuracy varies between 88.79% and 89.22% . We observe less than 1% of variation. The curve has no constant growth or decay but fluctuates depending on the value of P with a maximum level reached for P = 1200 giving 89.22% of mean accuracy.

Next, we fix the penalty parameter equal to 1200 and do vary the γ parameter in the range of 100 and 2000 with a step of $\frac{1}{10}$. As shown in Fig. 8(b), for γ in range 0.091 and 0.125, we observe constant accuracies. Then, the curve has a tendency to decrease as γ increases. This shows the important influence of γ on the results of the classification.



(a) Penalty (b) Gamma

Figure 8. SVM parameters optimization.

With the help of the previous study of the influence of different parameters, we set the penalty parameter to

1200 and γ to 0091 for our final classification. Fig. 9 illustrates the result of the classification on the study area in the north-east of the atoll of Vahitahi.

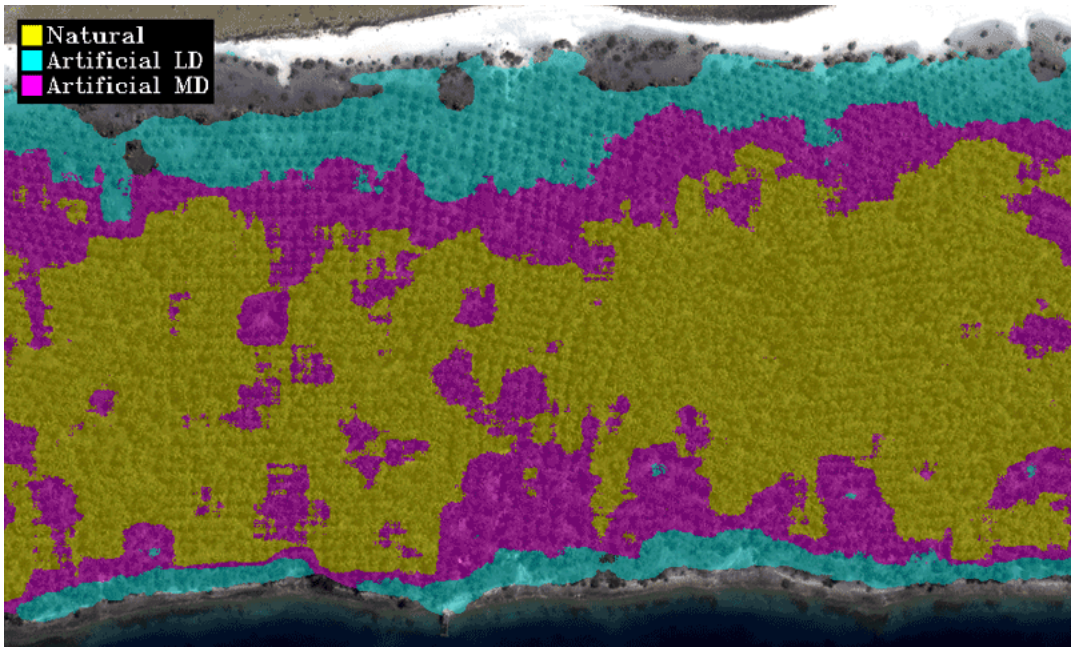


Figure 9. SVM classification using P=1200 and $\gamma = 0.091$.

6. CONCLUSION

In this study, we highlight that the use of spectral information not sufficient to classify the different field' types. But we reach our goal by providing textural informations. Thanks to the study of classifiers' accuracies, we show that the use of Minimun Distance and Parallelepiped algorithms produces poor results. Mahalanobis algorithm gives results that are satisfactory but not sufficient to classify correctly fields' types of coconut trees. Best results are obtained with Support Vector Machines and Maximum Likelihood classifications. But, through the study of the influence of various parameters of SVM and although Maximum Likelihood's accuracies are almost equivalent to that of SVM, we show that SVM gives very satisfactory results with a good selection of parameters P and γ.

The use of the SVM classification is a good tool to estimate from remote sensed images different fields type of coconut trees, and allows an estimation of the number of trees in each class according recommendations about the space between trees in artificial fields.

Because of the lack of the Near Infra-red channel, future works could focus on finding other descriptors based on vegetation indices calculated using the RGB channels or other color space descriptions to improve the quality of classification.

REFERENCES

[1] Freund, J. E., [*Mathematical Statistics*], Prentice Hall, Upper Saddle River, NJ, 5 ed. (1992).
[2] Cortes, C. and Vapnik, V., "Support-vector networks," *Machine Learning* **20**(3), 273–297 (1995).
[3] Burges, C. J. C., "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery* **2**(2), 121–167 (1998).
[4] Wu, T.-F., Lin, C.-J., and Weng, R. C., "Probability estimates for multi-class classification by pairwise coupling," *Journal of Machine Learning Research* **5**, 975–1005 (2004).
[5] Swain, P. and Davis, S., [*Remote Sensing, the Quantitative Approach*], Mc Graw-Hill, NY (1978).

[6] Wacker, A., *The Minimum Distance Approach to Classification*, PhD thesis, Purdue University, West Lafayette (1971).

[7] Kailath, T., "The divergence and bhattacharyya distance measures in signal selection," *Communications, IEEE Trans. Communication Theory* .

[8] Ricotta, C. and Avena, C., "The influence of principal component analysis on the spatial structure of a multispectral dataset," *International Journal of Remote Sensing* **20**(17), 3367–3376 (1999).

[9] Haralick, R. M., Dinstein, and Shanmugam, K., "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-3**, 610–621 (November 1973).

[10] Haralick, R., "Statistical and structural approaches to texture," *Proceedings of the IEEE* **67**, 786–804 (1979).

[11] Anys, H., Bannari, H., He, D., and Morin, D., "Texture analysis for the mapping of urban areas using airborne MEIS-II images," (3), 231–245 (1994).

[12] Puissant, A., Hirsch, J., and Weber, C., "The utility of texture analysis to improve per-pixel classification for high to very high spatial resolution imagery," *International Journal of Remote Sensing* **26**, 733–745 (February 2005).

[13] Chang, C.-C. and Lin, C.-J., *LIBSVM: a library for support vector machines* (2001). Software avaiable at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.