HAL
open science

Approche parallèle pour l'estimation du flot optique par méthode variationnelle

Riadh Fezzani

To cite this version:

Riadh Fezzani. Approche parallèle pour l'estimation du flot optique par méthode variationnelle. Traitement des images [eess.IV]. Université Pierre et Marie Curie - Paris VI, 2011. Français. NNT : . tel-00713970

HAL Id: tel-00713970

https://theses.hal.science/tel-00713970v1

Submitted on 3 Jul 2012

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

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

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PIERRE ET MARIE CURIE

École Doctorale de Sciences Mathématiques de Paris Centre

Spécialité

Mathématiques appliquées

Présentée par

M. Riadh Fezzani

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ PIERRE ET MARIE CURIE

Sujet de la thèse :

Approche parallèle pour l'estimation du flot optique par méthode variationnelle

Soutenue le 10 Juin 2011 *devant le jury composé de* :

Président du jury :	Frédéric Nataf	*Laboratoire J.L. Lions (UPMC)*
Rapporteur :	Zakaria Belhachmi	*Université de Haute Alsace*
Rapporteur :	Damien Tromeur-Dervout	*Université LYON 1*
Directeur de thèse :	François-Xavier Roux	*Laboratoire J.L. Lions (UPMC)*
Encadrant :	Frédéric Champagnat	*ONERA (The french aerospace Lab.)*
Examinateur :	Étienne Mémin	*INRIA (Rennes)*

Résumé :
Cette thèse s'inscrit dans le sillage des nombreuses recherches menées autour du problème d'estimation du flot optique. Elle se distingue par le fait d'avoir été menée dans un cadre pluridisciplinaire. Elle traite en effet d'un problème fondamental en traitement d'images (flot optique) auquel ont été appliqué des techniques de calcul scientifique (parallélisation) en visant une application expérimentale de la mécanique des fluides (vélocimétrie par images de particules).

Le flot optique désigne le champ de vitesses apparent observé entre deux images successives d'une même scène. Son estimation est un intermédiaire essentiel dans de très nombreuses applications et nous proposons d'appliquer des techniques venant de la communauté de la vision par ordinateur à la vélocimétrie par images de particules (PIV).

À la suite d'une évaluation préliminaire des approches classiques d'estimation du flot optique employant une régularisation soit locale soit globale du champ de vecteurs, nous décidons de considérer l'approche CLG de Bruhn *et al.* [2005b] qui propose de combiner les régularisations locale et globale. Nous étudions dans un premier temps la mise en œuvre du "*warping*" dans CLG puis nous le justifions dans notre approche baptisée "*modified CLG*" (MCLG).

Nous décrivons ensuite deux méthodes de résolution. La première repose sur une approche par calcul variationnel et permet le développement d'un solveur de type Picard-SOR par bloc employant uniquement des linéarisations locales. Ce solveur est parallélisé par une technique de "color labelling".
La deuxième méthode est une approche par "variable splitting" qui repose sur le fait d'introduire une seconde variable dans le modèle ce qui permet de le séparer en sous problèmes plus faciles à résoudre. Les solveurs intermédiaires employés dans cette techniques se composent d'opérations décorrélées, se qui facilite leur parallélisation.

Une évaluation complète des différentes méthodes mises en œuvre au cours de ces travaux est présentée. On s'intéresse particulièrement au parallélisme des algorithmes développés, et cela sur différentes architectures (CPU multi-cœurs et GPU). On mesure également leur précison à l'aide de séquences classiques de traitement d'images mais aussi à l'aide d'images de PIV. On montre ainsi l'efficacité de la parallélisation de ces algorithmes (aussi bien sur CPU que sur GPU) et on prouve également l'interrés de considérer une approche CLG en PIV.

Mots clés : Flot optique, calcul paralléle, PIV, CLG, warping.

Parallel approach for optical flow estimation by variational method.

Abstract :
This thesis deals with optical flow estimation. It was made in a multidisciplinary context. In fact, this work is the resolution of a fundamental problem in the image processing area (optical flow) by using scientific computing methods (parallel algorithms) to solve an experimental application from fluids mechanics area (Partical Image Velocimetry).

The optical flow is defined as the vector field discribing the apprent motion between two successive images. It is an important source of information, so that, it is used in different kind of applications. We propose here to apply some image processing tools to particle image velocimetry (PIV).

After a brief evaluation of the local and global regularization of vector field used for optical flow estimation, we consider the CLG method of Bruhn *et al.* [2005b] which proposes to combine these two kinds of regularizations. We first study the implementation of the warping in the large displacement contest and we justify it in our modified CLG (MCLG) approach.

We describe then two solvers. The first uses variational approaches to derive a bloc Picard-SOR solver only employing local linearisations. We also described a color labelling scheme allowing the parallelisation of this solver.
The second approach rely on a variable splitting technique which introduces an auxiliary variable in the considered model and to split the problem into two sub-problems easier to solve. This method only involve independent operations, its parallelisation is then simple.

A full evaluation of the developped algorithms is presented. We give particular attention to their parallel properties on different architectures (multicore CPU and GPU). We then study their precision using some classical image proccessing datas and also PIV datas.

Keywords : Optical flow, parallel computing, PIV, CLG, warping.

Remerciements

Ceux qui ont assisté à ma soutenance ont pu remarquer à quel point je suis maladroit pour faire des remerciements. Je vais tout de même essayer d'exprimer ici toute la gratitude que je ressens envers bon nombre de personnes.

Je tiens à remercier avant tout l'ensemble des membres du jury de ma thèse. Je remercie en particulier Frédéric Nataf qui m'a fait l'honneur d'accepter de présider le jury et avec qui je vais avoir le plaisir de collaborer dans le cadre d'un post-doc. Je remercie d'autre part les deux rapporteurs, Zakaria Belhachmi et Damien Tromeur-Dervout, du soin qu'ils ont apporté à la rédaction des rapports et pour les nombreuses observations et critiques éclairées qu'ils ont formulées à l'égard de mes travaux.
Mes remerciements vont également à Étienne Mémin, qui par son expertise dans le domaine du traitement d'image (particulièrement le calcul du flot optique appliqué aux images de fluides) représente à mes yeux une caution importante pour valoriser mes travaux.

Comment ne pas remercier également François-Xavier Roux, mon directeur de thèse, (FX pour les intimes...) qui permet à l'équipe CHP (Calul Hautes Performances) de travailler dans des conditions enviables. Son expérience de numéricien et sa perspicacité m'ont apporté certaines réponses précieuses. Sa bonne humeur et sa gentillesse ne sont plus à prouver.

En cherchant de l'inspiration sur internet [1], j'ai pris conscience de la chance que j'ai eu d'être encadré par Frédéric Champagnat et Guy Le Besnerais. Ce magnifique duo m'avait d'ailleur été vanté lors de mon premier entretien à la DRH de l'ONERA. Leur réputation n'est pas volée. Avec humour et bienveillance, ils sont toujours disponibles et de bon conseil. Leur aide m'a également été très pécieuse lors de la rédaction d'articles ou de mon manuscrit. Dans leur souci permanent de perfectionnement, ils essayent de mettre au point un numéro de bon et mauvais flics ☺...

Je regretterai cette chère équipe CHP au sein de laquelle j'ai passé trois riches années grâce à l'ensemble de ses membres et notamment Xavier Juvigny. Xavier, à l'esprit bouillonant, toujours disponible et pleins d'idées. Il m'a beaucoup aidé, à l'occasion de nos petits déjeuners rituels du jeudi, ou encore pendant des séances de debuggage intensives.

C'est au tour du DTIM (Département de Traitement de l'Information et Modélisation) auquel appartient CHP. J'y ai eu le plaisir de côtoyer plusieurs personnes qui ont fait de ma thèse une belle aventure. Je ne citerai pas tout le monde (je n'aurai pas assez de place ici...) mais j'insisterai quand même sur les adeptes de la "belette qui tète". Cependant, je tiens à adresser un petit message à Martial, notre

1. http ://findethese.blogspot.com

nounou à tous, toujours prêt à donner les encouragements nécessaires en période de stress. Je pense aussi à Mister GPU, Aurélien, qui m'a donné un coup de main précieux pour organiser le pot de thèse et avec qui on partage le duo d'encadrants de choc... Nos poses café "geeky" me manqueront sûrement.

Marine (ma petite "keub") a dû faire le plein de patience pour supporter mes humeurs. Je n'ai sûrement pas été très rigolo, surtout vers la fin (bon, d'accord, pas que vers la fin ☺). Merci pour tout.

Je ne dois pas non plus oublier mes parents. Ils sont sûrement pour quelque chose dans ce qui m'arrive ☺. Le soutien et l'amour inconditionnel, c'est quand même bien agréable. Merci.

Je finirai par une petite pensée pour Major Tom... A-t-il un tourne-disque dans son vaisseau ? Sa fin est-elle triste ou non ? Le mistère reste entier... Peut être qu'un autre passage au "sans sanz" pourra nous éclairer, n'est ce pas Guy ?..

Table des matières

1 Introduction **1**

1.1 Corrélation d'images ("block matching") 2

1.2 Introduction au calcul du flot optique 3

1.2.1 Contrainte fondamentale du flot optique 3

1.2.2 Formulations bien posées pour le calcul du flot optique 5

1.2.3 Mesure de grands déplacements 9

1.3 Calcul parallèle 10

1.3.1 État de l'art des méthodes parallèles pour calculer le flot optique 11

1.4 Vélocimétrie par images de particules 12

1.4.1 Dispositif d'acquisition 12

1.4.2 Traitement des images de particules 13

1.4.3 Flot optique appliqué à la PIV 15

1.5 Objectif et organisation 15

2 Combinaison des approches par régularisation locale et globale pour l'estimation du flot optique **19**

2.1 Justification intuitive de la combinaison des régularisations locale et globale 20

2.1.1 Étude d'images synthétiques de particules 20

2.1.2 Étude de données de PIV réelles 21

2.2 Méthode CLG (*combined local and global*) de Bruhn *et al.* 25

2.2.1 Gestion des grands déplacements 26

2.3 Justification de l'usage du warping dans CLG 26

2.3.1 Discussion de l'usage du warping dans CLG de Bruhn *et al.* . 26

2.3.2 MCLG ou la mise en œuvre efficace et correcte du warping dans CLG 28

2.4 Évaluation ampirique de la combinaison des méthodes locale et globale 28

2.5 Conclusion 29

3 Calcul variationnel **33**

3.1 Équations d'Euler-Lagrange 33

3.2 Discrétisation de la partie régularisation 35

3.2.1 Approche classique 35

3.2.2 Approche alternative 35

3.3 Élaboration du solveur non linéaire 37

3.3.1 Linéarisation locale des équations 38

3.3.2 Méthode SOR non linéaire par bloc 2×2 39

3.4 Parallélisation du solveur 40

3.4.1 Coloration de mailles "*Red-Black*" 41

3.4.2 Coloration de mailles à quatre couleurs 41

3.5 Conclusion 42

4 Résolution par la méthode de ("*variable splitting*") 43
4.1 Reformulation du problème de calcul du flot optique 43
4.2 Méthode de Bregman 45
4.3 Application à la méthode MCLG 47
4.3.1 Résolution de la partie adéquation aux données 47
4.3.2 Résolution de la partie régularisation 48
4.3.3 Parallélisation 49
4.4 Conclusion 49

5 Évaluation du parallélisme des méthodes développées 51
5.1 Détails de mise en œuvre 51
5.1.1 Schéma multi-niveaux 52
5.1.2 Construction des systèmes 53
5.1.3 Gestion des bords 53
5.1.4 Structure algorithmique 54
5.1.5 Décompte des opérations 54
5.2 Mise en œuvre CPU/OpenMP 56
5.2.1 Modèle de parallélisation utilisé 56
5.2.2 Sélection des versions comparées 56
5.2.3 Résultats 57
5.3 Mise en œuvre GPU/CUDA 63
5.3.1 Le modèle de programmation sur GPU 63
5.3.2 Mesure de performances 65
5.4 Conclusion 67

6 Évaluation de la précision des méthodes développées 69
6.1 Évaluation sur des séquences "classiques" 69
6.2 Évaluation et validation de l'application à la PIV 74
6.2.1 Résolution spatiale 76
6.2.2 Réponse au bruit 78
6.2.3 Réponse fréquentielle 83
6.2.4 Peak locking 86
6.3 Conclusion 86

7 Conclusion 91
7.1 Perspectives 93

A Schéma classique pour la discrétisation de la partie régularisation 95
A.1 Détail du développement classique 95
A.2 Dérivation du critère discret 96
A.2.1 Différence avant 97
A.2.2 Différence arrière 97
A.2.3 Combinaison des différences avant et arrière 98

A.3 Conclusion 98

B Coefficients des système locaux induit par les méthodes variationnelles du chapitre 3 99

B.1 Régularisation anisotrope 100

B.1.1 Méthode classique 100

B.1.2 Méthode alternative 101

B.2 Régularisation isotrope 101

B.2.1 Méthode classique 101

B.2.2 Méthode alternative 102

C Décomposition de domaine 103

C.1 Approche naïve 103

C.2 Méthode de Schwarz avec recouvrement 104

Liste des symboles 107

Bibliographie 109

Table des figures

1.1 Exemple de flot optique. **À gauche :** dixième image de la séquence rubberWhale de la collection Middlebury (disponible à l'adresse http ://vision.middlebury.edu/flow/data/). **Au centre :** flot optique de référence correspondant. **À droite :** code couleur utilisé pour représenter l'intensité et la direction des vecteurs. 3

1.2 Illustration de l'estimation de grands déplacements par la technique de warping. 10

1.3 Illustration des différentes techniques de parallélisation. **À gauche :** MIMD à mémoire partagée. **Au centre :** MIMD à mémoire distribuée. **À droite :** SIMD. 11

1.4 Dispositif d'acquisition d'images de particules. 13

1.5 Exemple d'images de particules : couple d'images tiré de la séquence A du PIV challenge 2003 [Stanislas *et al.*, 2005]. Noter la zone plus lumineuse présente sur l'image de droite qui peut perturber l'estimation du mouvement. 14

2.1 **À gauche :** première image de l'exemple de cas synthétique utilisé pour l'évaluation. Pour une meilleure qualité d'impression, le contraste et la luminosité de cette image ont été augmentés. **À droite :** le champ de vecteurs décrivant le déplacement exact des pixels de cette séquence ainsi que sa norme. Les couleurs varient de bleu (pas de déplacement) à rouge (déplacement de 7.26 pixels). Les carrés rouge et vert délimitent les zones étudiées dans les FIGURES 2.3 et 2.4. 21

2.2 **À gauche :** norme du résultat de la méthode par régularisation locale avec un rayon de fenêtre égal à 7 pixels. **À droite :** norme du résultat de la méthode par régularisation globale avec le paramètre α égal à 20. 22

2.3 Courbes de niveau de la norme sur la zone du carré rouge (*c.f.*FIGURE 2.1). **À gauche :** vérité terrain. **Au centre :** résultat de la méthode locale. **À droite :** résultat de la méthode globale. . . 22

2.4 Norme en relief de la zone du carré vert (*c.f.*FIGURE 2.1). **À gauche :** vérité terrain. **Au centre :** résultat de la méthode locale. **À droite :** résultat de la méthode globale. 22

2.5 Séquence A du PIV challenge 2001. Données réelles représentant un tourbillon. 23

2.6 **À gauche :** norme du résultat de la méthode par régularisation locale avec un rayon de fenêtre égal à 10 pixels. **À droite :** norme du résultat de la méthode par régularisation globale avec le paramètre α égal à 20. 23

2.7 Contours des normes sur la zone centrale des résultats de la FIGURE 2.6. **À gauche :** résultat de la méthode locale. **À droite :** résultat de la méthode globale. 24
2.8 Coupe horizontale à travers le cœur du tourbillon (le long de la ligne 333) de la composante verticale des résultats présentés dans la FIGURE 2.6. 24
2.9 Coupe horizontale à travers le cœur du tourbillon (le long de la ligne 333) de la composante verticale des résultats obtenus pour la méthode locale avec un rayon de fenêtre de 3 et la méthode globale avec $\alpha = 100$. 25
2.10 Différentes zones du domaine image Ω sur laquelle l'intensité lumineuse et ses dérivées sont évaluées dans les équations (2.6) et (2.8). **À gauche :** zone concernée à l'instant t. **Au centre :** zone à l'instant $t+1$ par un recalage de fenêtres correct (2.6). **À droite :** zone à l'instant $t+1$ considérée lors du warping tel que mis en œuvre par Bruhn *et al.* dans la méthode CLG (2.8). 27
2.11 Traitement du cas synthétique par la méthode MCLG avec un rayon de fenêtre de 3 et $\alpha = 60$ comparé avec les résultats des méthodes locale et globale présenté dans les FIGURES 2.2, 2.3 et 2.4. **(1) :** norme du résultat. **(2) :** Contour des normes sur la zone du carré rouge (*c.f.*FIGURE 2.1). **(3) :** Norme en relief de la zone du carré vert (*c.f.*FIGURE 2.1). **(a) :** résultats de la méthode locale. **(b) :** résultats de la méthode globale. **(c) :** résultats de la méthode MCLG. Le résultat de MCLG présente un compromis entre le respect des zones de fort gradient (image de contour deuxième ligne) et le lissage du bruit des zones de faible déplacement (surface 3D troisième ligne). . 30
2.12 Traitement du cas réel par la méthode MCLG avec un rayon de fenêtre de 3 et $\alpha = 60$, comparé avec les FIGURES 2.6, 2.7 et 2.8. **(1) :** norme du résultat. **(2) :** Contour des normes au niveau du cœur du tourbillon. **(3) :** Coupe horizontale à travers le cœur du tourbillon (le long de la ligne 333) de la composante verticale. **(a) :** résultats de la méthode locale. **(b) :** résultats de la méthode globale. **(c) :** résultats de la méthode MCLG. On voit la précision de la méthode MCLG au niveau du centre du tourbillon (*c.f.* (2)) ainsi que le lissage du bruit présents dans les zones de faible déplacement dans la solution de la méthode globale (*c.f.* (3) les oscillation de la méthode globale en s'éloignant du centre du tourbillon). 31

3.1 Coloriage des mailles avec deux couleurs. **À gauche :** pattern des interactions entre voisins. **À droite :** Grille coloriée. 41
3.2 Coloriage des mailles avec quatre couleurs. **À gauche :** pattern des interactions entre voisins. **À droite :** Grille coloriée. 41

5.1 Exemple de pyramide dyadique. 53
5.2 Schéma de résolution multi-niveau (ou multigrille unidirectionnel). . 53

5.3 Proportion des calculs effectués sur les grilles grossières dans un schéma multi-échelles unidirectionnel. Le nombre de niveaux de la pyramide correspond au maximum possible sous condition que la grille la plus grossière soit de dimension au moins 16×16. 54

5.4 Mesure des performances des algorithmes avec 12 threads et taille d'images variable (de 300×300 à 1900×1900). **Haut gauche :** temps de calcul. **Haut droit :** nombre d'opérations par seconde. **Bas gauche :** Accélération. **Bas droit :** Efficacité. 58

5.5 Mesure des performances des différentes méthodes développées en faisant varier le nombre de threads avec une dimension d'images fixe de 700×700. **Haut gauche :** efficacité. **Milieu gauche :** accélération. **Bas gauche :** temps de calcul. **droite :** nombre d'opérations flottantes par seconde. 60

5.6 Mesure des performances des différentes méthodes développées à charge de calcul fixe par thread (300×300 pixels). **Haut gauche :** efficacité. **Milieu gauche :** accélération. **Bas gauche :** temps de calcul. **droite :** nombre d'opérations flottantes par seconde. 62

5.7 Modèle de programmation sur GPU et présentation des différents types de mémoires et des manières d'y accéder. 64

5.8 Mesure des performances de la version GPU de l'algorithme $MCLG^{CV}_{ISO_{4ppv}}$ comparé à sa version CPU à taille d'images variable (de 300×300 à 1900×1900). Le code GPU emploie soit la mémoire globale soit la mémoire locale. **Haut gauche :** temps de calcul. **Bas gauche :** Accélération. **Droit :** Nombre d'opération par seconde. . . 66

6.1 Compilation des meilleurs résultats en terme d'erreur angulaire pour les sept algorithmes développés au cours de cette thèse sur les données de test de Middlebury. 71

6.2 Comparaison des méthodes de gestion de la partie régularisation sur le cas Urban3. **À gauche :** meilleur solution pour $MCLG^{CV}_{ISO_{8ppv}}$ (α=10, $\sigma = 0.66$). **À droite :** meilleur solution pour $MCLG^{CV}_{ISO_{4ppv}}$ (α=10, σ=1). 72

6.3 Résultats sur les quels les erreurs du tableau 6.1 sont mesurées. Ligne : **(1)** Dimetrodon, **(2)** Grove2, **(3)** Grove3, **(4)** Hydrangea, **(5)** RubberWhale, **(6)** Urban2, **(7)** Urban3, **(8)** Venus. Colonne : **(a)** Solution de référence. **(b)** $MCLG^{CV}_{8ppv_{ISO}}$. **(c)** $CLG^{CV}_{8ppv_{ISO}}$. **(c)** $MCLG^{VS}$. 73

6.4 Compilation des meilleurs résultats en terme d'erreur angulaire pour les différents algorithmes développés au cours de cette thèse sur le cas synthétique d'images de particules représentant une grille de tourbillons (*c.f.*FIGURE 2.1). 75

6.5 Norme du champ de vecteurs présentant une variation horizontale brusque de 0.25 à 0.35 pixel. Les zones encadrées sont celles étudiées dans la suite. **En orange :** zone de transition qui servira à évaluer la résolution. **En vert :** mouvement vertical uniforme, à gauche, déplacement de 0.25 pixel, à droite, déplacement de 0.35 pixel. On évaluera sur cette zone la réponse au bruit. 75

6.6 Profil moyen du déplacement horizontal pour le cas de la FIGURE 6.5. **Gauche :** Les solutions de la méthode par régularisation locale avec un rayon d efenêtre R=1, 5 et 31. **Droite :** Les solutions de la méthode par régularisation globale avec paramètre de régularisation α=2, 10 et 100. 76

6.7 **À gauche :** Résolution spatiale de la méthode par régularisation locale en fonction du diamètre de la fenêtre. **À droite :** Résolution spatiale de de la méthode globale en fonction du paramètre de régularisation. 77

6.8 Résolution spatiale de la méthode MCLG avec différents rayons de fenêtre d'interrogation en fonction du paramètre de régularisation α. 77

6.9 Les différents paramètres des méthodes locale, globale et MCLG en fonction de la résolution mesurée. Ces courbes permettent d'identifier l'influence de ces paramètres sur la résolution. 78

6.10 Agrandissement d'une même zone d'une image de la séquence FIGURE 6.5 dans les différentes conditions considérées. **De gauche à droite :** cas 0,1,2 et 3. 79

6.11 **À gauche :** Erreurs absolues moyennes mesurées en fonction du rayon de la fenêtre d'interrogation de la méthode locale sur les quatre cas étudiés. **À droite :** Erreurs absolues moyennes mesurées en fonction du paramètre de régularisation α de la méthode globale sur les quatre cas étudiés. 79

6.12 Erreurs absolues moyennes mesurées en fonction du paramètre de régularisation de MCLG avec différents écarts types de fenêtre gaussienne σ. **Haut gauche :** cas 0. **Haut droit :** cas 1. **Bas gauche :** cas 2. **Bas droit :** cas 3. 80

6.13 Erreurs absolues moyennes mesurées en fonction de la résolution des méthodes locale, globale et MCLG avec différents paramètres. **Haut gauche :** cas 0. **Haut droit :** cas 1. **Bas gauche :** cas 2. **Bas droit :** cas 3. 81

6.14 Comparaison des profils moyens des solutions de MCLG, méthode locale et méthode globale pour le meilleur compromis bruit/résolution. **Haut gauche :** cas 0. **Haut droit :** cas 1. **Bas gauche :** cas 2. **Bas droit :** cas 3. 82

6.15 Norme du champ de vecteurs représentant un cisaillement sinusoïdal d'amplitude 2 pixels et de longueur d'onde 200 pixels. 83

6.16 Profil moyen du déplacement horizontal pour le le cas du cisaillement sinusoïdal de longueur d'onde 100 pixels. **En haut :** Les solutions de la méthode par régularisation locale avec un rayon R=5 et 31. **En bas :** Les solutions de la méthode par régularisation globale avec paramètre de régularisation α=10 et 90. 84
6.17 Ratio entre l'amplitude de la solution et l'amplitude réelle du mouvement en fonction de la résolution normalisée. **À gauche :** méthode locale avec différents rayons de fenètre. **À droite :** méthode globale en fonction de différents paramètres de régularisation. 85
6.18 Ratio entre l'amplitude de la solution de MCLG et l'amplitude réelle du mouvement en fonction de la résolution normalisée. Les résultats de différents écarts types de fenêtre gaussienne σ sont présentés avec des paramètres de régularisation α fixés. 85
6.19 Séquence représentant le cisaillement sinusoïdal : profil moyen de la solution de la méthode par régularisation locale comparé à celui de la méthode MCLG pour différentes longueurs d'ondes (de haut en bas, l= 60, 100, 200, 400). 87
6.20 Flot optique calculé à l'aide de MCLG de la séquence A du PIV challenge 2003 (*c.f.*FIGURE 1.5) ainsi que sa norme. **À gauche :** Déplacement instantané correspondant à la première paire d'images. **À droit :** Déplacement moyen mesuré sur les 100 paires d'images. . 88
6.21 On cherche à observer le peak locking sur l'ensemble de la séquence A du PIV challenge 2001. **À gauche :** Histogramme du déplacement horizontal. **À droit :** Histogramme des déplacements fractionnaires. 88

C.1 Interface entre deux sous-domaines Ω_1 et Ω_2 et illustration des communications nécessaires. 104
C.2 Illustration de la méthode de Schwarz avec recouvrement. 105

Chapitre 1

Introduction

Sommaire

1.1 Corrélation d'images ("block matching") 2
1.2 Introduction au calcul du flot optique 3
1.2.1 Contrainte fondamentale du flot optique 3
1.2.2 Formulations bien posées pour le calcul du flot optique 5
1.2.3 Mesure de grands déplacements 9
1.3 Calcul parallèle 10
1.3.1 État de l'art des méthodes parallèles pour calculer le flot optique 11
1.4 Vélocimétrie par images de particules 12
1.4.1 Dispositif d'acquisition 12
1.4.2 Traitement des images de particules 13
1.4.3 Flot optique appliqué à la PIV 15
1.5 Objectif et organisation 15

Ce travail s'inscrit dans le vaste domaine de l'estimation du mouvement apparent dans une séquence d'images.
Le mouvement en traitement d'image est une riche source d'informations. C'est en effet un intermédiaire nécessaire à un certain nombre d'applications.
Décrire les déplacements dans une séquence d'images peut être utilisé par exemple à des fins d'observation : détection de mouvement pour la surveillance, estimation de la densité du trafic routier, comptage du nombre de passants, suivi d'objets... Le mouvement dans ces exemples provient du déplacement des objets observés.
Le mouvement peut également donner une information sur la structure de la scène capturée. En robotique par exemple, on utilise le mouvement observé à l'aide d'une caméra embarquée pour établir une carte de profondeur. On est ainsi capable de détecter automatiquement les obstacles et donc d'éviter les collisions. Dans ce cas-ci, le mouvement apparent dans la séquence est produit par le déplacement de la caméra.
Enfin, le déplacement mesuré à un certain instant dans une séquence d'images peut être utilisé pour "prédire" les images suivantes. Ce principe est la base de la compression vidéo par compensation de mouvement (MPEG).

Des applications à vocation plus fondamentale existent néanmoins. L'observation en mécanique étant un préalable à la compréhension des phénomènes physiques, on retrouve dès le début du vingtième siècle des expériences basées sur l'observation

du déplacement d'un fluide.
Ludwig Prandtl est le précurseur de telles techniques [Raffel *et al.*, 2007]. Son expérience s'appuie sur l'observation de particules emportées par un fluide en mouvement dans un canal. Son procédé permet de décrire qualitativement les caractéristiques du déplacement du fluide.
La vélocimétrie par images de particules (PIV) correspond à la réalisation moderne de cette expérience. La PIV a connu un grand essor depuis les années 1980. Cette application fournit un cadre expérimental important pour les méthodes d'estimation de mouvement qui sont étudiées dans ce travail. Elle pose aussi des problèmes spécifiques qui seront présentées plus tard dans ce chapitre.

Les sections qui suivent présentent les principales techniques de détermination du mouvement apparent dans une séquence d'images.

1.1 Corrélation d'images ("block matching")

Estimer le mouvement entre deux images revient à faire correspondre les éléments de la première avec ceux de la deuxième. On pourra les lier ainsi par un champ de vecteurs.
Le procédé intuitif consiste à découper la première image en zones plus ou moins grandes. Pour chacune de ces imagettes (appelées aussi zones d'intérêt ou encore fenêtres d'interrogation), on parcourt la seconde image (la totalité ou une partie) à la recherche de la zone qui lui ressemble le plus. Pour cela, on résout le problème d'optimisation

$$\arg\max_{\delta x, \delta y} \int_{\mathcal{V}(x,y)} I(x' + \delta x, y' + \delta y, t+1) I(x', y', t) dx' dy'. \tag{1.1}$$

où $I : \Omega \times \mathbb{R} \to \mathbb{R}$ désigne l'intensité d'un point P, de coordonnées cartésiennes (x, y), dans le domaine image $\Omega \subset \mathbb{R}^2$. $\mathcal{V}(x, y) \subset \Omega$ désigne un voisinage de (x, y) et $(\delta x, \delta y)$ est le vecteur décrivant le déplacement de (x, y).

Le compromis classique associé à ce procédé est le suivant : d'une part, plus les imagettes sont de taille réduite, plus le nombre de positions dans la seconde image susceptibles d'être solution augmente. Le problème devient mal posé.
D'autre part, si la dimension des imagettes est trop importante, les structures fines du déplacement ne peuvent pas être restituées. On perd ainsi en précision.
Finalement, le nombre d'opérations dépend à la fois de la taille de la zone d'intérêt et de la taille de la zone de recherche dans la seconde image. Si ces dimensions sont trop importantes, les temps de calcul deviennent prohibitifs. On passe alors en général dans le domaine fréquentiel en calculant la corrélation (1.1) par une FFT pour accélérer les temps de calcul.

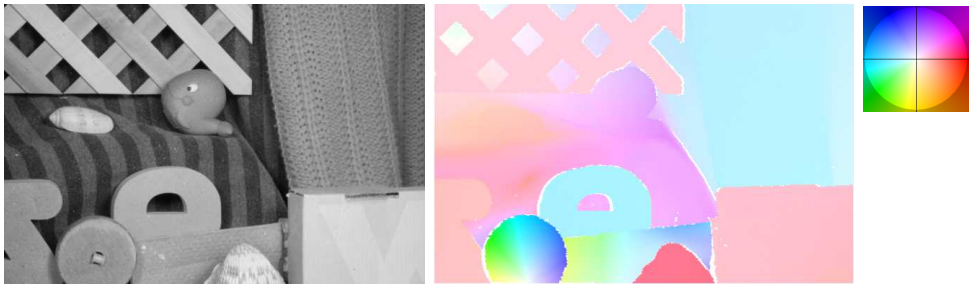

FIGURE 1.1 – Exemple de flot optique. **À gauche :** dixième image de la séquence rubberWhale de la collection Middlebury (disponible à l'adresse http ://vision.middlebury.edu/flow/data/). **Au centre :** flot optique de référence correspondant. **À droite :** code couleur utilisé pour représenter l'intensité et la direction des vecteurs.

1.2 Introduction au calcul du flot optique

Le terme de flot optique désigne le champ de vecteurs décrivant le mouvement de chaque pixel dans une séquence d'images. Contrairement à la méthode de corrélation d'images dans laquelle on associe un vecteur par bloc de pixels, on cherche ici à produire un vecteur par pixel. La FIGURE 1.1 est un exemple de représentation du flot optique par un code de couleur permettant d'indiquer la direction et l'intensité du déplacement de chaque pixel de la séquence. Cet exemple est tiré des données d'évaluation fournie par l'université de Middlebury [Baker *et al.*, 2009].

Pour calculer le flot optique, on suppose qu'un pixel conserve son intensité lumineuse au cours de son déplacement. Cette condition est implicite dans les méthodes par corrélation de fenêtres dans lesquelles on suppose qu'une zone d'intérêt (caractérisée par l'intensité des pixels qui la composent) de la première image est présente dans la deuxième image.

La condition de conservation de l'intensité lumineuse peut s'écrire

$$I(x+\delta x, y+\delta y, t+\delta t) - I(x,y,t) = 0 \qquad (x,y) \in \Omega; \quad t \in \mathbb{R}, \tag{1.2}$$

δx et δy sont les coordonnées cartésiennes du vecteur associé à P décrivant son déplacement au cours de l'intervalle de temps δt. Elles dépendent de la position (x, y) du point considéré ainsi que de l'instant t auquel elles sont mesurées.

Remarque 1.2.1 *Pour une écriture plus légère de la condition de conservation de l'intensité, on utilise plusieurs abus de notation. En effet x et y dépendent du temps et devraient s'écrire $x(t)$ et $y(t)$. D'autre part, δx et δy dépendent des coordonnées du point auquel ils sont considérés, ainsi que du temps. On devrait donc utiliser $\delta x(x,y,t)$ et $\delta y(x,y,t)$.*

1.2.1 Contrainte fondamentale du flot optique

Horn et Schunck [1981] ainsi que Lucas et Kanade [1981] dans leurs travaux

fondateurs utilisent une version approchée de la condition de conservation de l'intensité (1.2). En dérivant l'intensité lumineuse $I(x(t), y(t), t)$ par rapport au temps, on obtient :

$$\frac{\partial I}{\partial t} + \frac{\partial I}{\partial x} \cdot \frac{dx}{dt} + \frac{\partial I}{\partial y} \cdot \frac{dy}{dt} = 0. \tag{1.3}$$

En identifiant $(\frac{\delta x}{\delta t}, \frac{\delta y}{\delta t})$ avec $(\frac{dx}{dt}, \frac{dy}{dt})$ (la vitesse du point (x, y)), on définit la contrainte fondamentale du flot optique :

$$\partial_t I + \partial_x I \cdot u + \partial_y I \cdot v = 0, \tag{1.4}$$

où $\partial_* I = \frac{\partial I}{\partial *}$ et $(u, v) = (\frac{\delta x}{\delta t}, \frac{\delta y}{\delta t})$ désigne le flot optique recherché.

Remarque 1.2.2 *La dérivée partielle de l'intensité par rapport au temps est obtenue par différence finie*

$$\partial_t I = \frac{I(x, y, t + \delta t) - I(x, y, t)}{\delta t}. \tag{1.5}$$

Un usage assez répandu en pratique est de prendre $\delta t = 1$, c. à d. *qu'on utilise la cadence d'acquisition des images comme unité de temps. De cette manière, $(u, v) = (\delta x, \delta y)$. On utilisera dans la suite $\delta t = 1$.*

La contrainte fondamentale seule ne permet pas de calculer le flot optique. En effet, ne disposant que d'une unique équation (1.4) pour deux inconnues (u et v), on obtient une infinité de solutions possibles. Cette sous-détermination est désignée par Horn et Schunck [1981] sous le nom de "problème d'ouverture" (*aperture problem*).

Remarque 1.2.3 *On peut résoudre l'indétermination de la contrainte fondamentale du flot optique* (1.4) *en la dérivant par rapport à x puis par rapport à y, on obtient le système linéaire suivant :*

$$\begin{pmatrix} \partial^2_{xx} I & \partial^2_{xy} I \\ \partial^2_{xy} I & \partial^2_{yy} I \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \partial^2_{xt} I \\ \partial^2_{yt} I \end{pmatrix}. \tag{1.6}$$

On note ici $\partial^2_{**} I = \frac{\partial^2 I}{\partial * \partial \star}$.

On retrouve ici la méthode de Tretiak et Pastor [1984] ou encore celle de Tistarelli [2002] qui ont supposé la conservation du gradient spatial de l'intensité lumineuse $\nabla I(x, y, t) = (\partial_x I, \partial_y I)$. Cette approche a le défaut d'être sensible au bruit, parce qu'elle fait intervenir les dérivées au second ordre de l'intensité.

1.2.2 Formulations bien posées pour le calcul du flot optique

On trouve dans la riche bibliographie qui traite du problème du flot optique différentes solutions permettant d'obtenir un système bien posé à partir de la contrainte fondamentale du flot optique (1.4) [Aubert et Kornprobst, 2006].

On notera dans ce qui suit

$$\mathbf{x} = (x, y, t) \tag{1.7}$$

$$\mathbf{u} = (u, v, 1) \tag{1.8}$$

1.2.2.1 Régularité locale

On peut supposer, par continuité, que deux points voisins dans une image ont un déplacement proche. On peut imposer cette contrainte au mouvement estimé en supposant que le mouvement dans une séquence d'images est constant par morceaux [Lucas et Kanade, 1981]. Cela revient à dire que les pixels situés dans un même voisinage se meuvent solidairement.

Remarque 1.2.4 *Cette condition peut s'avérer fausse, notamment si le voisinage considéré comporte des pixels se trouvant de part et d'autre d'une frontière.*

Avec la condition de régularité locale, on ramène le problème de calcul du flot optique à la minimisation au sens des moindres carrés de

$$\int_{\Omega} \left(\int_{\mathcal{V}(x,y)} \left(I(\mathbf{x}' + \mathbf{u}(\mathbf{x})) - I(\mathbf{x}') \right)^2 dx'dy' \right) dxdy, \tag{1.9}$$

où $\mathbf{x}' = (x', y', 1)$ avec $(x', y') \in \mathcal{V}(x, y)$.

En remplaçant dans cette équation la condition de conservation de l'intensité (1.2) par la contrainte fondamentale du flot optique (1.4) et en remarquant que l'équation précédente est séparable en tout $\mathbf{x}$, la minimisation de l'énergie globale (1.9) revient à minimiser pour chaque point $(x, y) \in \Omega$ l'expression suivante

$$\int_{\mathcal{V}(x,y)} \left(\partial_x I(\mathbf{x}') \cdot u(\mathbf{x}) + \partial_y I(\mathbf{x}') \cdot v(\mathbf{x}) + \partial_t I(\mathbf{x}') \right)^2 dx'dy'. \tag{1.10}$$

Pour cela, on cherche à annuler le gradient de (1.10) par rapport à (u, v). Le système linéaire qui en résulte est donc

$$\begin{pmatrix} \int_{\mathcal{V}(x,y)} (\partial_x I)^2 & \int_{\mathcal{V}(x,y)} \partial_x I \partial_y I \\ \int_{\mathcal{V}(x,y)} \partial_x I \partial_y I & \int_{\mathcal{V}(x,y)} (\partial_y I)^2 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \int_{\mathcal{V}(x,y)} \partial_x I \partial_t I \\ \int_{\mathcal{V}(x,y)} \partial_y I \partial_t I \end{pmatrix}. \tag{1.11}$$

Remarque 1.2.5 *On peut aussi choisir de minimiser l'énergie globale* (1.9) *en commençant par annuler sa dérivée en tout point* $(x, y) \in \Omega$ *par rapport à* $\mathbf{u}(\mathbf{x})$*. On obtient alors*

$$\int_{\mathcal{V}(x,y)} \nabla^T I(\mathbf{x}' + \mathbf{u}(\mathbf{x})) \left(I(\mathbf{x}' + \mathbf{u}(\mathbf{x})) - I(\mathbf{x}')\right) dx'dy' = 0. \tag{1.12}$$

En remplaçant ensuite la condition de conservation de l'intensité (1.2) *par la contrainte fondamentale du flot optique* (1.4)*, on retrouve le système linéaire* (1.11)*.*

Remarque 1.2.6 *La sommation sur un voisinage* $\mathcal{V}$ *de* $(x, y) \in \Omega$ *dans* (1.10) *peut s'écrire à l'aide d'une opération de convolution. En effet, en définissant*

– $\mathcal{U}$ *la fonction indicatrice de* $\mathcal{V}(x, y)$ *:*

$$\mathcal{U}(x', y') = 1 \qquad \text{si } (x', y') \in \mathcal{V}(x, y), \tag{1.13}$$

$$= 0 \qquad \text{sinon.} \tag{1.14}$$

– $\nabla_3 I = (\partial_x I, \partial_y I, \partial_t I)$,

L'expression à minimiser (1.10) *se note*

$$\mathcal{U} * \left(\mathbf{u}(\mathbf{x}) \nabla_3 I^T \nabla_3 I \mathbf{u}^T(\mathbf{x})\right) \tag{1.15}$$

Cette formulation peut être modifiée en remplaçant la fonction indicatrice $\mathcal{U}$ *par une gaussienne* $\mathcal{G}$ *centrée en* (x, y)*. Ce choix se justifie du fait que l'influence* (x, y) *sur le déplacement de* $(x', y') \in \Omega$ *diminue à mesure que la distance entre eux est grande. En d'autres termes, plus un point est éloigné de* (x, y)*, plus la contrainte de régularité locale est fausse, elle doit donc être relâchée. L'emploi de la gaussienne* $\mathcal{G}$ *permet de donner un poids plus important aux points proches de* (x, y)*. Plutôt que l'énergie* (1.15)*, on considère alors*

$$\mathcal{G} * \left(\mathbf{u}(\mathbf{x}) \nabla_3 I^T \nabla_3 I \mathbf{u}^T(\mathbf{x})\right) \tag{1.16}$$

La classe des méthodes par régularisation locale est connue pour sa robustesse au bruit présent dans les données traitées. Le temps de calcul est également un avantage, en effet le nombre d'opérations nécessaires est limité :

– calcul et multiplication des dérivées partielles $\partial_x I, \partial_y I$ et $\partial_t I$, puis leur sommation sur le voisinage $\mathcal{V}(x, y)$ pour la construction des systèmes linéaires locaux
– résolution d'un système 2×2 par pixel.

En revanche, les méthodes locales souffrent de leur manque de précision. En effet, l'hypothèse de constance par morceaux ne permet pas de mesurer certains phénomènes locaux, typiquement, lorsque leur échelle spatiale est inférieure à la taille de la fenêtre. On est donc susceptible de perdre les caractéristiques locales du mouvement.

Remarque 1.2.7 *On peut considérer que les méthodes de corrélations s'apparentent à l'estimation du flot optique par approche locale car, on suppose dans les deux cas que tous les pixels d'une fenêtre d'interrogation ont le même déplacement. Une étude de comparaison de ces deux approches est menée par Champagnat et al. [2011] sur des données de vélocimétrie par image de particules : leurs comportements sont très similaires.*
Une relation explicite existe entre la mesure de corrélation et la condition de régularité locale modélisée par l'équation (1.9)*. En effet, minimiser*

$$\int_{\mathcal{V}(x,y)} \left(I(\mathbf{x}' + \mathbf{u}(\mathbf{x})) - I(\mathbf{x}')\right)^2 dx'dy' \tag{1.17}$$

par rapport à $\mathbf{u}(\mathbf{x})$ *est équivalent à la minimisation de*

$$\int_{\mathcal{V}(x,y)} I(\mathbf{x}' + \mathbf{u}(\mathbf{x}))^2 - I(\mathbf{x}' + \mathbf{u}(\mathbf{x}))I(\mathbf{x}')dx'dy'. \tag{1.18}$$

On voit apparaître dans la deuxième partie de cette expression le produit de corrélation entre les images considérées. L'équivalence entre les deux méthodes est montrée sous certaines conditions par Heitz et al. [2010].

1.2.2.2 Régularité globale

Horn et Schunck [1981] ont adopté une approche différente qui a inspiré depuis un très grand nombre de méthodes.
Plutôt que d'imposer une régularité locale au champ de vecteurs, on suppose qu'il est globalement régulier, par exemple qu'il est continûment dérivable par morceaux. On peut par exemple pénaliser les gradients de la solution souhaitée.
Le problème s'écrit alors comme la minimisation d'une fonctionnelle sous la forme :

$$\int_\Omega \underbrace{\mathcal{D}(I(\mathbf{x}+\mathbf{u}), I(\mathbf{x}))}_{\text{Adéquation aux données}} + \alpha \underbrace{\mathcal{R}(\nabla u, \nabla v)}_{\text{régularisation}} dxdy, \tag{1.19}$$

où $\alpha \in \mathbb{R}$ est un paramètre positif permettant de donner plus ou moins de poids au terme de régularisation de l'énergie. $\mathcal{D}$ et $\mathcal{R}$ sont en général des fonctions convexes et agissent respectivement sur le terme dit d'adéquation aux données ou sur le terme de régularisation du champ de vecteurs. Ces approches correspondent à appliquer une régularisation de Tikhonov [1943] au problème de l'estimation du flot optique. Une vaste bibliographie discute du choix de ces deux fonctions de pénalisation et de la manière d'optimiser le critère (1.19). Horn et Schunck [1981] ont considéré une pénalisation quadratique dans les deux parties de la fonctionnelle :

$$\int_\Omega (\partial_x I \cdot u + \partial_y I \cdot v + \partial_t I)^2 + \alpha(|\nabla u|^2 + |\nabla v|^2)dxdy. \tag{1.20}$$

Cette formulation a l'avantage de fournir un système linéaire lors du passage aux équations d'Euler-Lagrange [Cartan, 1977]. Pour le résoudre ils ont choisi d'utiliser la méthode de Jacobi par blocs [Horn et Schunck, 1981].

Le choix d'une fonction quadratiques pour $\mathcal{R}$ a, en revanche, le désavantage de lisser la solution de manière uniforme, alors que dans beaucoup d'applications, le calcul d'un flot optique précis passe par une bonne restitution des discontinuités du mouvement.
Une stratégie consiste à considérer que les discontinuités du mouvement sont liées aux contours d'intensité présents dans les images. On pondère alors la régularisation quadratique en fonction du gradient spatial des images étudiées, ce procédé est souvent appelé régularisation "*image driven*". Alvarez *et al.* [1999], dans cet esprit, proposent la fonctionnelle de régularisation

$$\mathcal{R}_1 = \phi(|\nabla I|)(|\nabla u|^2 + |\nabla v|^2) \tag{1.21}$$

où ϕ est une fonction positive décroissante.
La régularisation de Alvarez *et al.* applique le même poids aux deux composantes du flot, ce qui correspond à une régularisation dite isotrope. Nagel et Enkelmann [1986] préconisent pour leur part de lisser dans la direction de la tangente aux discontinuités de l'intensité des images mais pas dans la direction orthogonale à celles-ci. Pour cela, ils utilisent le terme de régularisation suivant :

$$\mathcal{R}_2 = \nabla u^T \mathcal{M}(\nabla I)\nabla u + \nabla v^T \mathcal{M}(\nabla I)\nabla v, \tag{1.22}$$

où

$$\mathcal{M} = \frac{1}{|\nabla I|^2 + 2\varepsilon}\begin{pmatrix} \partial_y I^2 + \varepsilon^2 & -\partial_x I \partial_y I \\ -\partial_x I \partial_y I & \partial_x I^2 + \varepsilon^2 \end{pmatrix} \tag{1.23}$$

avec ε assez petit.

Une autre approche, proposée par Black [1992] consiste à remplacer le critère quadratique initialement utilisé par Horn et Schunck [1981] par une fonction pénalisante robuste qui pénalise moins les fortes discontinuités qu'une fonction quadratique :

$$\mathcal{R}_3 = \phi(|\nabla u|, |\nabla v|). \tag{1.24}$$

De nombreux auteurs préconisent la minimisation de la variation totale du champ de vecteur [Cohen, 1993; Kumar *et al.*, 1996; Papenberg *et al.*, 2006; Zach *et al.*, 2007]. L'utilisation de la variation totale pour le respect des discontinuités des solutions a été introduite par Rudin, Osher, et Fatemi [1992] pour le problème du débruitage d'images.
La variation totale d'une fonction $f \in L^1(\Omega)$ ayant un gradient $\nabla f \in L^1(\Omega)$ est définie par

$$TV(f) = \int_\Omega |\nabla f(x,y)| dxdy \tag{1.25}$$

Le terme de régularisation (1.24) emploie dans ce cas la fonction $\phi(x,y) = x + y$.
Enfin, Aubert *et al.* [1999]; Aubert et Kornprobst [1999] de même que Blanc-Feraud *et al.* [1993] ont utilisé des fonctions inspirées de l'estimation statistique robuste telle que la fonction quadratique tronquée.

Avec le même souci de robustesse aux données aberrantes Black et Anandan [1996] préconisent le choix d'une fonction de pénalisation $\mathcal{D}$ non quadratique. Ce type de pénalisation est appliqué depuis par une grande majorité des publications traitant du sujet. Ces données aberrantes correspondent en fait aux zones d'exception à la contrainte de conservation de l'intensité lumineuse, typiquement, les zones d'occlusion ou encore les zones affectées par des variations d'éclairage...

1.2.3 Mesure de grands déplacements

Une limite dans la formulation de Horn et Schunk (1.20) et dans celle de Lucas et Kanade (1.10) est le fait qu'elles ne sont correctes que dans le cadre des petits mouvements.
En effet, la contrainte fondamentale du flot optique (1.4) peut être obtenue grâce à un développement de Taylor :

$$I(x+u, y+v, t+1) = I(x, y, t+1) + \partial_x I \cdot u + \partial_y I \cdot v \tag{1.26}$$
$$= I(x, y, t) \qquad \text{d'après (1.2).} \tag{1.27}$$

$\partial_t I$ étant en pratique égal à $I(x, y, t+1) - I(x, y, t)$ d'après (1.5), on retrouve bien la contrainte fondamentale du flot optique (1.4). Or, le développement limité (1.26) n'est correct que si le déplacement (u, v) est assez petit. Beaucoup de recherches ont été faites pour permettre d'appréhender de grands mouvements.

En considérant le problème à plusieurs résolutions et en partant du constat que, sur les niveaux les plus grossiers, l'échelle des déplacements est plus petite (donc que le développement (1.26) est valide), la stratégie classique (dite "*coarse to fine*") consiste à commencer le traitement à une échelle grossière et à résoudre le problème sur des grilles de plus en plus fines. La solution d'un niveau au suivant et utilisée pour compenser le mouvement entre les deux images et initialiser le nouveau problème [Anandan, 1989; Black et Anandan, 1991] (voir la FIGURE 1.2.3). Cette technique est connue dans la littérature sous le nom de "*warping*" [Papenberg *et al.*, 2006].

Plus précisément, le warping consiste à itérativement répéter le développement (1.26) en introduisant une séquence $(\mathbf{u}^n)_{n\in\mathbb{N}}$ définie par

$$\mathbf{u}^n = (u^n, v^n, 1), \tag{1.28}$$
$$\mathbf{u}^{n+1} = \mathbf{u}^n + \delta\mathbf{u}^n \qquad \text{où} \quad \delta\mathbf{u}^n = (\delta u^n, \delta v^n, 0). \tag{1.29}$$

On obtient alors

$$I(\mathbf{x} + \mathbf{u}^{n+1}) = I(\mathbf{x} + \mathbf{u}^n + \delta\mathbf{u}^n), \tag{1.30}$$
$$= I(\mathbf{x} + \mathbf{u}^n) + \partial_x I(\mathbf{x} + \mathbf{u}^n) \cdot \delta u^n + \partial_y I(\mathbf{x} + \mathbf{u}^n) \cdot \delta v^n, \tag{1.31}$$
$$= I(\mathbf{x}). \tag{1.32}$$

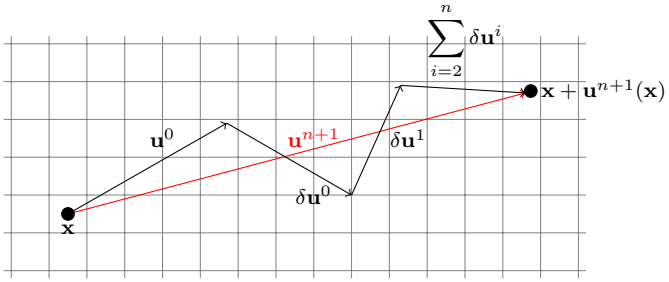

FIGURE 1.2 – Illustration de l'estimation de grands déplacements par la technique de warping.

Le Besnerais et Champagnat [2005] ont justifié et formalisé ce procédé dans le cadre de la régularisation locale. Papenberg *et al.* [2006] a fait de même dans le cadre de la régularisation globale.

1.3 Calcul parallèle

La puissance du matériel informatique a considérablement évolué au court des dernières décennies. Leur exploitation repose essentiellement sur le paradigme simple qui propose de partager la tâche à effectuer entre plusieurs unités de calcul (UC) et de les faire travailler en même temps ou en d'autres termes, en parallèle.

L'exploitation de ces architectures nécessite d'adapter les algorithmes afin de les décomposer en tâches indépendantes pouvant être effectuées en parallèle. Cependant, tous les algorithmes ne ce prêtent pas à ce genre de décomposition.

Plusieurs techniques de parallélisation existent et chacune se destine à une certaine architecture informatique (*c.f.*FIGURE 1.3) :

SIMD (Single Instruction, Multiple data), une seule instruction sera effectuée plusieurs fois en parallèle sur plusieurs données différentes. Il s'agit typiquement de l'architecture des processeurs graphiques (GPU) qui appliquent la même opérations en chaque pixel. Les machines vectorielles peuvent s'apparenter à ce fonctionnement, mais dans ce cas là, les opérations de calcul se recouvrent dans le temps par un système de pipeline.

MIMD (Multiple Instruction, Multiple data) ce modèle correspond à un parallélisme à grain grossier. Il est moins restrictif que le précédent car il permet de lancer différentes opérations sur plusieurs données. Deux choix de mise en œuvre se posent ici :

mémoire partagée les données traitées sont partagées par l'ensemble des unités de calcul, comme pour un processeur multi-cœurs par exemple,

mémoire distribuée chaque unité de calcul dispose de son propre espace de stockage, comme par exemple dans un cluster. Les unités de calcul peuvent si nécessaire s'échanger des données. Ce modèle est utilisé en

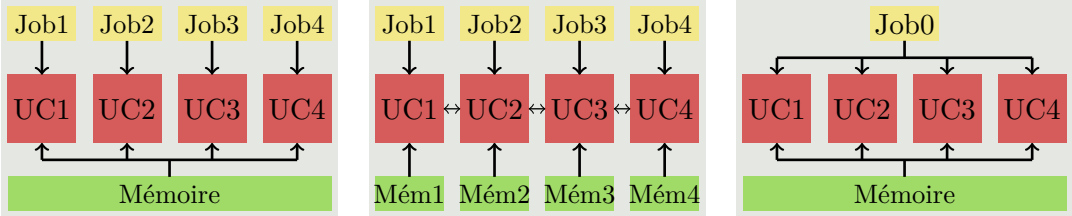

FIGURE 1.3 – Illustration des différentes techniques de parallélisation. **À gauche :** MIMD à mémoire partagée. **Au centre :** MIMD à mémoire distribuée. **À droite :** SIMD.

général lorsque la quantité de données à traiter est trop importante pour pouvoir être contenue dans un même espace mémoire.

On distinguera également deux types de parallélisation : celle à grain fin et celle à grain grossier. Dans la première, les calculs parallèles se font sur des petites quantités de données alors que dans la seconde, les traitements se font par paquet de données.

Remarque 1.3.1 *Une combinaison de ces techniques est possible. Par exemple, sur un cluster de processeurs multi-cœurs (ou de cartes graphiques), on pourrait envisager un partage des calculs en MIMD à mémoire distribuée entre chaque processeur qui, eux, effectueraient leurs tâches propres en SIMD.*

1.3.1 État de l'art des méthodes parallèles pour calculer le flot optique

L'accélération des algorithmes de calcul de flot optique est une exigence importante des méthodes actuelles. En effet, certaines applications, telle que la robotique, nécessitent des calculs à haute cadence.
D'autres domaines, comme la vélocimétrie par images de particules, traitent des images de plusieurs millions de pixels, ce qui entraîne que certaines expériences peuvent durer plusieurs heures. La réduction des temps de calculs est un enjeu considérable.
Kohlberger *et al.* [2005] ont développé une méthode de décomposition de domaine afin de paralléliser la résolution du modèle global pour le problème du flot optique. Cette approche permet une parallélisation selon le modèle MIMD à mémoire distribuée. Le stage de fin d'étude de Penel [2007], préliminaire à ces travaux de thèse, a également exploré cette voie de parallélisation.

Plus récemment, un certain nombre de méthodes ont été développées sur carte graphique, *c. à d.* selon le modèle parallèle SIMD. En ce qui concerne la classe des approches locales, l'algorithme de Champagnat *et al.* [2009], se décomposant en séquences d'opérations de convolution et de résolutions de système locaux, permet des calculs de flot optique à cadence vidéo sur des images de grande taille (de

l'ordre de 2000 × 2000 pixels).
Pour les approches globales, les premières méthodes portées sur GPU [Zach *et al.*, 2007; Werlberger *et al.*, 2009] ont introduit une variable auxiliaire afin de découpler les parties adéquation aux données et régularisation du modèle qu'ils considéraient. La première partie étant traitée par des opérations de seuillage et la seconde par une approche primal-dual [Chambolle, 2004].
Plus récemment, Gwosdek *et al.* [2010] ont développé une approche basée sur la méthode FED (*Fast Explicit Diffusion*) [Grewenig *et al.*, 2010] pour résoudre un modèle global de flot optique par calcul variationnel.

En plus d'une mise en œuvre plus simple, les méthodes s'appuyant sur un parallélisme à grain fin citées précédemment fournissent des temps de calcul plus faible par rapport aux méthodes MIMD de Kohlberger *et al.* [2005] et Penel [2007].

1.4 Vélocimétrie par images de particules

L'expérience de Prandtl évoquée en introduction de ce chapitre consiste à visualiser le mouvement d'un fluide en y incorporant des particules. La vélocimétrie par images de particules (PIV) s'appuie sur ce principe et constitue un vaste domaine expérimental regroupant l'imagerie plane classique (2C-PIV), la micro-PIV pour l'observation de phénomènes microscopiques [Bayt et Breuer, 2000], la stéréo-PIV (3C-PIV) permettant de reconstruire un champ de vecteurs à 3 composantes par stéréo-vision [Westerweel et Nieuwstadt, 1991] ou encore la PIV tomographique (3D-PIV) permettant de calculer le déplacement sur un volume du fluide par tomographie [Elsinga *et al.*, 2006].

La PIV s'appuie d'une part sur un dispositif sophistiqué d'acquisition d'images et d'autre part sur des techniques de traitement de ces images [Raffel *et al.*, 2007; Brossard *et al.*, 2009].

1.4.1 Dispositif d'acquisition

La Figure 1.4 schématise le dispositif de base de l'acquisition d'images de particules. Ses composantes principales sont :

un fluide ensemencé : les particules utilisées doivent être assez légères pour ne pas modifier les caractéristiques physiques du fluide, mais leur nombre et leur finesse doit permettre l'observation correcte du mouvement,

un laser : la capture des images se fait sur une coupe fine du fluide. Une même tranche du fluide étudié est éclairée à deux reprises par deux plans laser superposés à un intervalle de temps très court (5-10ns), cela afin de bien "figer" les particules lors de la capture et ainsi éviter l'apparition d'éventuelles traînées.

une caméra : l'acquisition des images digitales de particules se fait en général à l'aide d'une caméra à capteur CCD (*Charge Coupled Device*). Le déclenche-

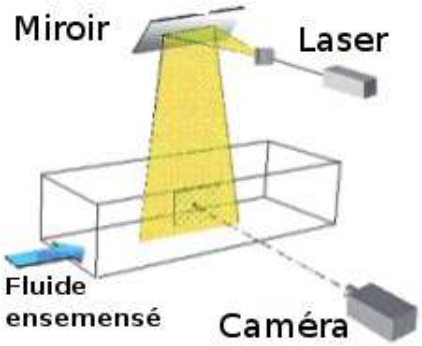

FIGURE 1.4 – Dispositif d'acquisition d'images de particules.

ment de la capture du couple d'images doit être très précisément synchronisé avec les deux impulsions laser. Les caméras utilisées aujourd'hui permettent d'acquérir des images haute définition (plusieurs millions de pixels).

Remarque 1.4.1 *Si l'intervalle de temps séparant la capture de deux images successives est très court en PIV classique, la fréquence d'acquisition de couple d'images est lui, en général, de l'ordre de 10Hz. Cette fréquence tend cependant à considérablement augmenter avec l'apparition de système de PIV "time resolved" (TR) permettant d'observer la corrélation temporelle des champs de vitesses de fluides.*

Le dispositif décrit précédemment est composé d'un matériel de pointe, très coûteux, destiné à produire un environnement contrôlé dans lequel tous les paramètres sont maîtrisés, notamment les dimensions de la surface capturée et l'intervalle de temps séparant l'acquisition de deux images successives. Cela afin de garantir une mesure précise du mouvement des fluides.

1.4.2 Traitement des images de particules

La démarche classique pour la détermination du mouvement en PIV consiste à procéder par corrélation d'images (section 1.1). On associe ainsi un vecteur par fenêtre d'interrogation. Des zones d'interrogations adjacentes peuvent se superposer, on utilise en général un recouvrement de 50%. Ceci permet de doubler le nombre de vecteurs mesurés dans chaque direction du domaine.

Plusieurs facteurs expérimentaux peuvent compromettre la condition de conservation de l'intensité lumineuse et donc détériorer la qualité du champ de vecteurs calculé :

bruit : la capture d'image est généralement entachée de bruit gaussien,

accident d'acquisition : des problèmes techniques (décalage des nappes lasers...) peuvent entraîner certaines anomalies telles que des variations d'illumination ou la présence de reflets,

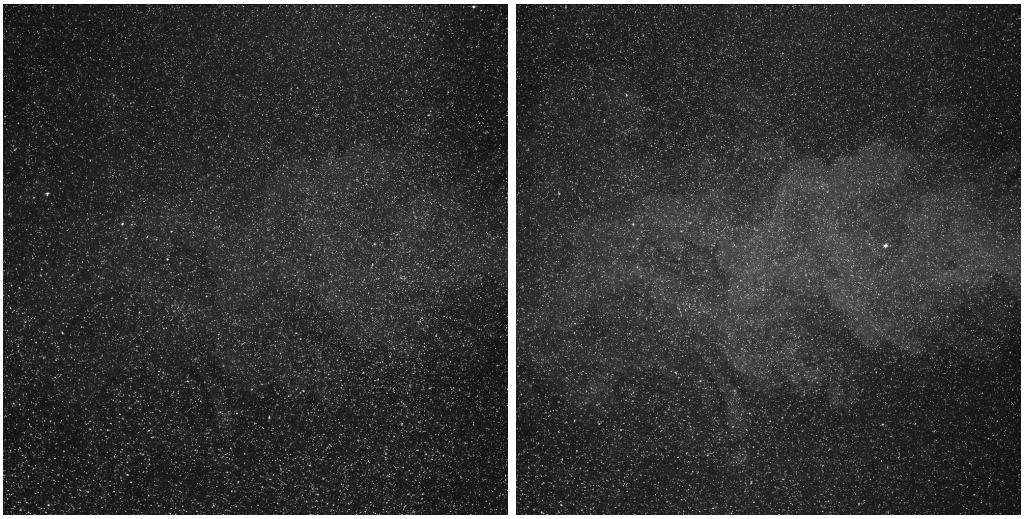

FIGURE 1.5 – Exemple d'images de particules : couple d'images tiré de la séquence A du PIV challenge 2003 [Stanislas *et al.*, 2005]. Noter la zone plus lumineuse présente sur l'image de droite qui peut perturber l'estimation du mouvement.

mouvement hors plan : si une partie du déplacement du fluide se fait perpendiculairement au plan de capture, un certain nombre de particules peut disparaître et d'autres apparaître entre la première et la deuxième image. Ce phénomène conduit à des exceptions locales à l'hypothèse de conservation de l'intensité lumineuse,

densité de particules : un nombre de particules capturées trop faible ne permet pas de caractériser précisément le mouvement du fluide. Malgré les efforts des expérimentateurs, cette densité ne peut pas être contrôlée partout. Par exemple, lors de la formation d'un tourbillon, les particules situées au niveau de son centre son éloignées (*c.f.*FIGURE 2.5),

taille des particules : lorsque la taille des particules est inférieure à 2 pixels, des effets de repliement spectral ("*aliasing*") peuvent limiter la précision subpixellique de l'estimation. On observe alors une tendance des déplacements vers des valeurs entières (en nombre de pixels). Ce phénomène est connu sous le terme de "*peak locking*" [Raffel *et al.*, 2007; Yamamoto et Uemura, 2009].

La FIGURE 1.5 est un exemple de données réelles tiré du PIV challenge 2003 [Stanislas *et al.*, 2005] qui illustre les difficultés rencontrées lors du traitement d'images de particules. Cette séquence représente un jet turbulent horizontal de droite à gauche. Cette séquence comporte un halo de fumée qui produit des variations locales de l'intensité lumineuse. D'autre part, les particules sont de petite taille, ce qui permet de mettre en évidence la sensibilité des techniques de calcul au peak locking.

Pour contrecarrer les difficultés rencontré lors de l'estimation du mouvement, on procède en général à une opération de validation ou de rejet de chaque vecteur estimé. Des vecteurs aberrants peuvent en effet se glisser dans le champ calculé.

Pour rejeter ces anomalies, on peut s'appuyer sur l'histogramme global du résultats ou encore appliquer différents filtres locaux (moyen, médian, différentiels...).

1.4.3 Flot optique appliqué à la PIV

Le terme "flot optique" est associé en PIV aux travaux de Quenot *et al.* [1998] qui utilisent de la corrélation de fenêtre et de la programmation dynamique. Bien que cette méthode produise un champ de vecteurs dense, elle est différente des techniques classiques de mesure du flot optique en traitement d'images qui ont été présentées précédemment.

Champagnat *et al.* [2009] ont appliqué avec succès une méthode de flot optique à régularisation locale au problème de vélocimétrie par images de particules. Cet algorithme hautement parallélisable a été porté sur carte graphique, ce qui lui permet de traiter plusieurs dizaines de paires d'images par secondes.

Ruhnau *et al.* [2005] montrent l'intérêt en terme de précision que peut représenter l'approche de Horn et Schunck appliquée à la PIV et développent à la suite de cela un terme de régularisation global basé sur les équations de Stokes [Ruhnau et Schnörr, 2007; Ruhnau *et al.*, 2007].

Corpetti *et al.* [2002, 2006] quant à eux ont adopté une approche globale en employant un terme de régularisation adapté aux mouvements de fluides. Plutôt que de pénaliser le gradient spatial du champ de vecteurs, ils préconisent d'agir sur sa divergence et son rotationel, selon le modèle de régularisation "Div-Curl" introduite par Suter [2002] en imagerie biomédicale.

Heitz *et al.* [2008] proposent de combiner une méthodes de corrélation à la méthode de Corpetti *et al.* [2002]. Cette approche peut être considéré comme une régularisation et une densification à posteriori du résultat de corrélation de fenêtres. En effet la solution de ce procédé associe un vecteur par pixel.

Remarque 1.4.2 *Heitz* et al. *[2010] proposent une revue des approches variationnelles appliquées à la mesure des déplacements de fluides et procèdent à une analyse de leurs performances. Les résultats qu'ils obtiennent montrent l'intérêt d'adapter les techniques de flot optique aux besoins de la PIV.*

1.5 Objectif et organisation

À cause de leur parenté, la corrélation d'images et les méthodes par régularisation locale souffrent des même défauts : ces deux approches, en considérant des zones contiguës relativement grandes pour caractériser le déplacement entre deux images, perdent les variations du déplacement à petite échelle que peuvent éventuellement contenir ces imagettes. En d'autres termes, l'hypothèse de constance par morceaux de la solution de ces méthodes est trop contraignante.
Cependant, les méthodes de flot optique (par régularisation locale ou globale) produisent un champ de vecteurs dense, ce qui est un avantage pour le traitement

des images de particules car cela permet d'accéder à certains détails du mouvement de fluides [Champagnat *et al.*, 2011].
En suivant les résultats de Ruhnau *et al.* [2005], le choix du type de méthode à employer se porte sur celles par régularisation globale. Néanmoins, bien que produisant des résultats de meilleure qualité que les méthodes par corrélation de fenêtres [Ruhnau *et al.*, 2005], le coût prohibitif des méthodes par régularisation globale non parallélisées les rend peu adaptées aux applications réelles de PIV où elles sont plutôt utilisées comme outils de validation [Quenot *et al.*, 1998].

Les travaux présentés dans ce manuscrit proposent d'élaborer une méthode rapide de calcul du flot optique par régularisation globale bien adaptée à la vélocimétrie par images de particules.
Pour cela, il est nécessaire de bien définir le modèle utilisé, et donc de caractériser le terme d'adéquation aux données et le type de régularisation choisi.
On prendra soin lors de l'élaboration des méthodes de résolution de développer des algorithmes permettant un traitement parallèle. La parallélisation à grain fin est favorisée, car, en accord avec la remarque 1.3.1, il sera possible d'inclure un tel algorithme dans un schéma parallèle à grain plus grossier comme le montrera l'annexe C.

Le chapitre 2 commence par l'examen de la qualité des résultats des méthodes de régularisation locale et ceux des méthodes de régularisation globale sur des images de particules. L'analyse qualitative qui suit suggère que l'approche de Bruhn *et al.* [2005b] consistant à combiner les deux types de régularisation est une méthode de flot optique adaptée à la PIV.
L'étude approfondie de cette méthode permet d'exhiber une approximation dans le traitement de la partie adéquation aux données de l'approche de Bruhn *et al.* [2005b]. Une formulation alternative pour combiner les régularisations locale et globale est alors proposée afin de corriger cette approximation. Deux méthodes de résolution sont proposées par la suite.

Le chapitre 3 s'intéresse pour sa part à la résolution par calcul variationnel du modèle proposé. Le modèle général considéré dans cette thèse est défini. Différentes manières de traiter la partie régularisation sont développées. Les détails de la discrétisation utilisée sont donnés ainsi que les différentes linéarisations employées pour appliquer une méthode de relaxation.
Pour permettre la parallélisation en SIMD de cette méthode, une technique de coloration de mailles (ou de réordonnancement des opérations) est utilisée afin de décorréler les opérations des solveurs décrits.

Dans le chapitre 4, le problème est considéré d'un autre point de vue. La méthode "*Split Bregman*" est définie dans le cadre général des méthodes de flot optique par régularisation globale.
Le choix d'utiliser cette méthode plutôt que celle de "*variable splitting*" par pénalisation quadratique (comme effectué par Zach *et al.* [2007] ou encore Werlberger

et al. [2009]) est justifié.
On détaillera ensuite l'application de la méthode développée à la résolution du modèle de flot optique considéré dans cette thèse.

L'étude se poursuit en évaluant les performances des solveurs décrits précédemment aussi bien en termes d'efficacité (temps de calcul) qu'en termes de précision des résultats fournis.
Au chapitre 5, les différents détails de mise en œuvre sont d'abord listés et justifiés. Les codes développés sont ensuite testés, d'une part, sur des multiprocesseurs disposant jusqu'à 12 unités de calcul, et d'autre part, sur des processeurs graphiques. Ces expériences cherchent à évaluer l'accélération apportée par les algorithmes construits au fil des chapitre 2, 3 et 4.
L'évaluation se poursuit au chapitre 6 : les erreurs sont mesurées sur des séquences de traitement d'images classiques issue de la base de données de l'université de Middlebury [Baker *et al.*, 2009] pour lesquelles une solution de référence est connue. Cette évaluation permet de se positionner par rapport à l'état de l'art des méthodes de flot optique en vision par ordinateur.
Pour clore ces chapitres d'expérimentations, des images de particules (aussi bien de synthèse qu'expérimentales) sont utilisées pour définir les caractéristiques de l'approche combinant les régularisations locale et globale. L'orientation choisie au chapitre 2 est alors confirmée.

En conclusion, un bilan des objectifs des travaux effectués au cours de cette thèse est fourni au chapitre 7 ainsi qu'une discussion des perspectives et voies de recherches qui restent ouvertes à l'issue de ce travail.

Chapitre 2

Combinaison des approches par régularisation locale et globale pour l'estimation du flot optique

Sommaire

2.1 Justification intuitive de la combinaison des régularisations locale et globale . . . 20
2.1.1 Étude d'images synthétiques de particules . . . 20
2.1.2 Étude de données de PIV réelles . . . 21
2.2 Méthode CLG (*combined local and global*) de Bruhn *et al.* . . . 25
2.2.1 Gestion des grands déplacements . . . 26
2.3 Justification de l'usage du warping dans CLG . . . 26
2.3.1 Discussion de l'usage du warping dans CLG de Bruhn *et al.* . . . 26
2.3.2 MCLG ou la mise en œuvre efficace et correcte du warping dans CLG . . . 28
2.4 Évaluation ampirique de la combinaison des méthodes locale et globale . . . 28
2.5 Conclusion . . . 29

Ce chapitre s'intéresse à la formulation du problème du flot optique en combinant les régularisations locale et globale. L'intérêt que suscite cette approche sera justifié dans un premier temps par la comparaison des résultats de la régularisation locale seule puis de la régularisation globale seule appliquées à des images de particules (section 2.1).
À la suite de cette courte analyse, la méthode "*combined local global*" (CLG) de Bruhn *et al.* [2005b] est introduite comme étant potentiellement la méthode permettant d'obtenir le meilleur des deux types de régularisation.

Le positionnement dans le contexte de mesure de grands mouvements, *c. à d.* en utilisant la contrainte non linéarisée de conservation de l'intensité lumineuse (1.2) (plutôt que l'utilisation de la contrainte fondamentale du flot optique (1.4) comme dans la version initiale de CLG [Bruhn *et al.*, 2005b]) permet d'exhiber une approximation dans la formulation de CLG proposée par Bruhn *et al.* Une solution est alors présentée pour éviter cette approximation.

Ce développement peut être considéré comme une justification théorique de la technique du "warping" pour la méthode CLG.

2.1 Justification intuitive de la combinaison des régularisations locale et globale

Pour justifier l'intérêt d'appliquer la méthode CLG au traitement d'images de particules, les deux classes de méthodes d'estimation du flot optique (locale et globale) seront testées. Un algorithme de chaque classe ayant déjà été appliqué à la vélocimétrie par images de particules est utilisé :

- FOLKI pour représenter les méthodes locales [Champagnat *et al.*, 2009],
- régularisation quadratique pour représenter les méthodes globales [Ruhnau *et al.*, 2005].

Ces deux méthodes seront évaluées dans cette partie sur deux jeux de données :

- une paire d'images synthétiques de particules (voir la FIGURE 2.1). Ces images nous ont été fournies par Benjamin Leclaire du département d'aérodynamique fondamentale et expérimentale (DAFE/ONERA). Elles sont produites à l'aide du générateur EUROPIV [Lecordier et Westerweel, 2004] et représentent une grille de trente tourbillons ayant différents diamètres et cinématiques.
- une paire d'images (*c.f.*FIGURE 2.5) tirée du PIV challenge 2001 [Stanislas *et al.*, 2003]. Ces données sont expérimentales et représentent un tourbillon. On ne dispose évidemment pas de vérité terrain pour cet exemple.

L'étude menée dans cette partie se concentre sur des exemples de tourbillons car ce phénomène physique est assez simple pour permettre des interprétations mais présente tout de même certaines difficultés qui illustrent bien les caractéristiques des différentes méthodes que l'on souhaite évaluer.

Les résultats présentés par la suite sont obtenus avec les meilleurs réglages possible pour les deux méthodes.

2.1.1 Étude d'images synthétiques de particules

On commence par montrer une vue globale des résultats obtenus au cours de cette première expérience. La norme des résultats des deux méthodes étudiées est présentée à la FIGURE 2.2. La solution de la méthode locale est visiblement plus lisse que celle de la méthode globale. Ce phénomène est particulièrement remarquable dans les zones ne comportant aucun déplacement (c'est à dire les zones bleues). Cependant, au niveau des tourbillons, c'est la régularisation globale qui donne apparemment le meilleur rendu. En effet, les limites de ces zones sont beaucoup plus régulières dans ce cas là que dans celui de la régularisation locale.

Pour mieux observer ces deux caractéristiques, un grossissement de ces zones d'intérêt est présenté aux FIGURES 2.3 et 2.4.

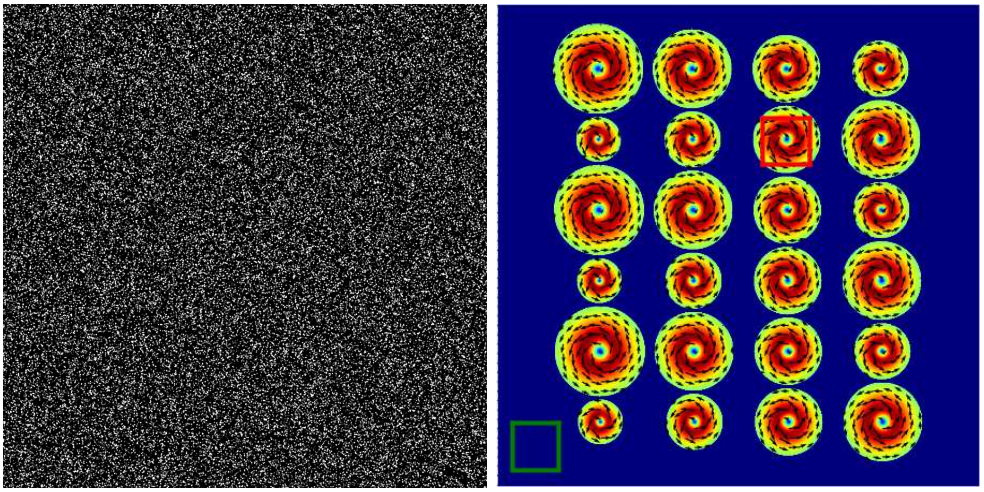

FIGURE 2.1 – **À gauche :** première image de l'exemple de cas synthétique utilisé pour l'évaluation. Pour une meilleure qualité d'impression, le contraste et la luminosité de cette image ont été augmentés. **À droite :** le champ de vecteurs décrivant le déplacement exact des pixels de cette séquence ainsi que sa norme. Les couleurs varient de bleu (pas de déplacement) à rouge (déplacement de 7.26 pixels). Les carrés rouge et vert délimitent les zones étudiées dans les FIGURES 2.3 et 2.4.

Pour commencer, la FIGURE 2.3 représente un gros plan sur la zone du septième tourbillon de la séquence étudiée (délimitée par un carré rouge dans la FIGURE 2.1). Le résultat de la méthode locale est nettement moins efficace que la méthode globale dans cette zone pour restituer la structure du tourbillon. En effet, là où la vérité terrain est représentée par des cercles concentriques, la solution de la régularisation locale est nettement moins fidèle à cette structure concentrique que celle de la régularisation globale.

Ensuite, la FIGURE 2.4 montre la représentation 3D de la norme dans la zone délimitée par le carré vert dans la FIGURE 2.1. Cette partie du domaine ne présente aucun mouvement, la vérité terrain est donc représentée par le plan $z = 0$. Cette fois-ci, on remarque la présence d'un bruit dans le résultat de la méthode globale qui n'est pas présent dans la solution de la méthode locale.

Cette expérience montre la capacité de la régularisation globale de conserver les structures fines du mouvement du fluide au sacrifice d'une certaine robustesse dans les zones présentant peu de déplacement. L'approche locale ne souffre pas de ce défaut, mais perd en précision, notamment au niveau du centre du tourbillon et au niveau de transition entre zones sans déplacement et le tourbillon.

2.1.2 Étude de données de PIV réelles

Le comportement observé sur les données synthétiques se confirme sur les données expérimentales. En effet, la FIGURE 2.6 montre que le résultat de la méthode locale est d'une manière générale plus lisse que celui de la méthode globale notam-

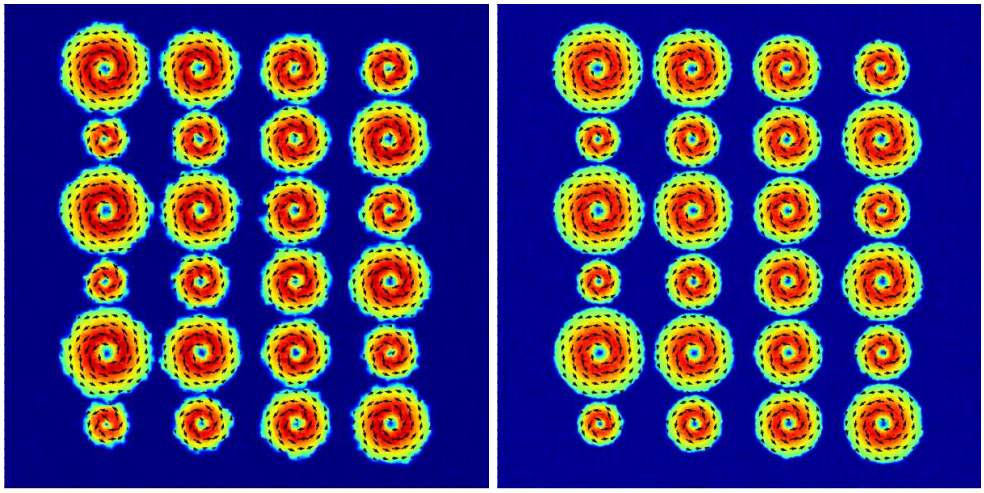

FIGURE 2.2 – **À gauche :** norme du résultat de la méthode par régularisation locale avec un rayon de fenêtre égal à 7 pixels. **À droite :** norme du résultat de la méthode par régularisation globale avec le paramètre α égal à 20.

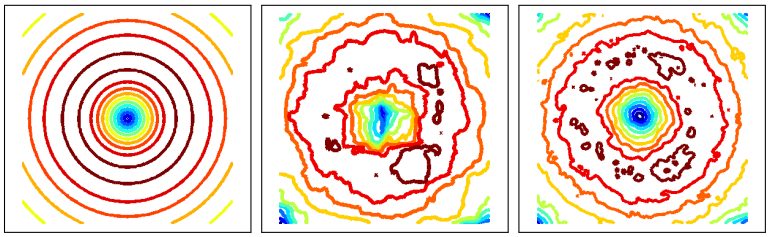

FIGURE 2.3 – Courbes de niveau de la norme sur la zone du carré rouge (*c.f.*FIGURE 2.1). **À gauche :** vérité terrain. **Au centre :** résultat de la méthode locale. **À droite :** résultat de la méthode globale.

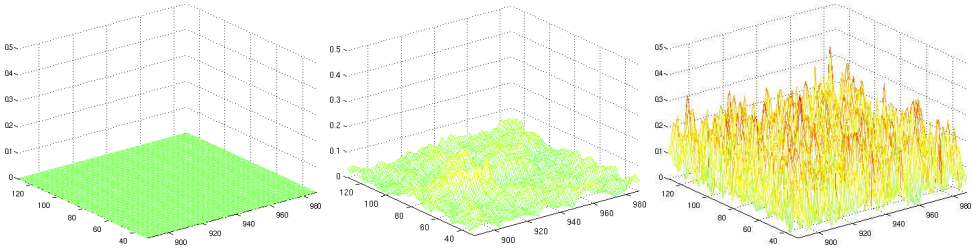

FIGURE 2.4 – Norme en relief de la zone du carré vert (*c.f.*FIGURE 2.1). **À gauche :** vérité terrain. **Au centre :** résultat de la méthode locale. **À droite :** résultat de la méthode globale.

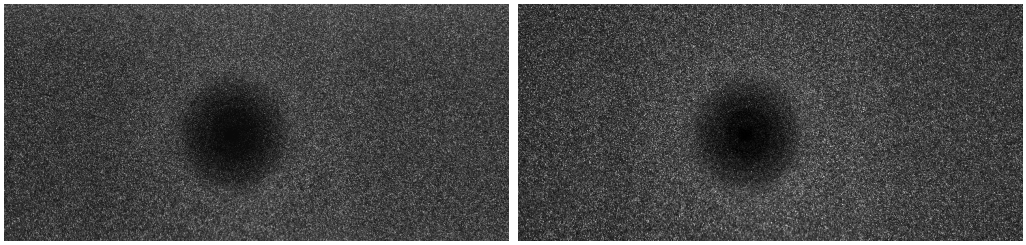

FIGURE 2.5 – Séquence A du PIV challenge 2001. Données réelles représentant un tourbillon.

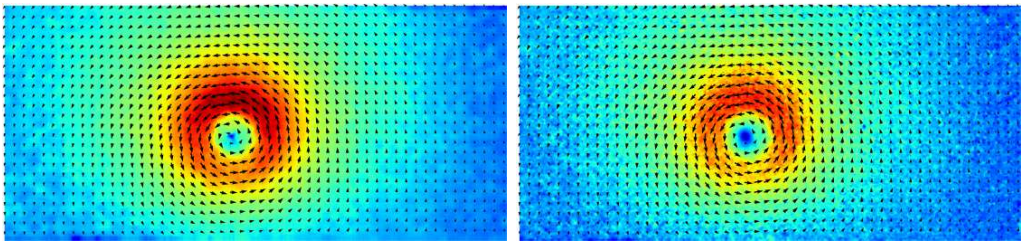

FIGURE 2.6 – **À gauche :** norme du résultat de la méthode par régularisation locale avec un rayon de fenêtre égal à 10 pixels. **À droite :** norme du résultat de la méthode par régularisation globale avec le paramètre α égal à 20.

ment dans les zones comportant un déplacement faible (les zones bleues).
On voit cependant, dans le cas de la régularisation globale, une transition au centre du tourbillon allant du bleu (déplacement de faible intensité) au rouge (grands mouvements), alors que dans cette zone, la méthode locale ne présente quasiment pas d'annulation du mouvement (couleur bleue) ce qui est pourtant une caractéristique du centre du tourbillon.
Pour confirmer cette observation, un agrandissement de la zone tourbillonnaire dans la FIGURE 2.7 montre que le résultat de la méthode globale, bien que plus chahuté que celui de la méthode locale, reproduit des courbes de niveau plus circulaires et concentriques.
Ainsi, l'approche globale retrouve mieux les propriétés physiques du tourbillon (simulées précédemment dans les données de synthèse).

Une autre manière de différencier les propriétés des approches locale et globale est de représenter une coupe horizontale de la composante verticale du champ de vecteurs le long d'une ligne passant à travers l'œil du tourbillon. Les courbes de la FIGURE 2.8 sont le résultat des deux méthodes testées.
L'hypothèse de régularité locale produit une courbe lisse mais avec la particularité de présenter une "marche" au niveau du cœur du tourbillon. On passe ainsi rapidement de valeurs négatives à des valeurs positives.
L'hypothèse de régularité globale quant à elle produit une transition plus douce entre les valeurs négatives et positives de la courbe, ce qui est un résultat plus en accord avec la modélisation physique. En revanche, on observe que plus on s'éloigne

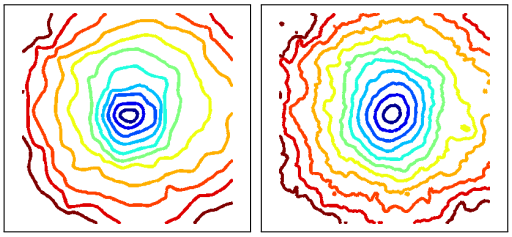

FIGURE 2.7 – Contours des normes sur la zone centrale des résultats de la FIGURE 2.6. **À gauche :** résultat de la méthode locale. **À droite :** résultat de la méthode globale.

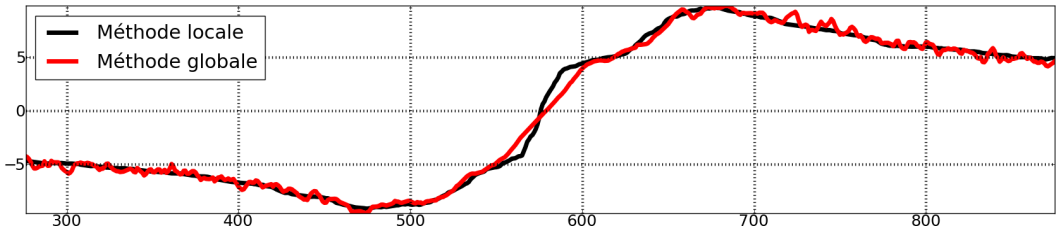

FIGURE 2.8 – Coupe horizontale à travers le cœur du tourbillon (le long de la ligne 333) de la composante verticale des résultats présentés dans la FIGURE 2.6.

du cœur du tourbillon, plus la courbe oscille. Ceci est à rapprocher des observations faites au niveau des zones homogènes dans la FIGURE 2.4 car plus on s'éloigne du centre du tourbillon plus la vitesse est faible. On retrouve alors les caractéristiques des méthodes observées dans les zones de faible vélocité sur l'exemple d'images synthétiques.

Remarque 2.1.1 *Au cœur du tourbillon, la force centrifuge repousse les particules, ce qui produit une zone sombre (voir* FIGURE *2.5). Ne disposant pas d'informations, seule la régularisation du champ de vecteurs permet de définir le déplacement dans ces zones. Or, la régularisation locale suppose que la solution est constante par morceaux alors que la régularisation globale, elle, suppose la continue différentiabilité par morceaux. D'où la différence des comportements.*

Pour corriger les défauts des deux méthodes, on serait tenté d'agir sur leurs paramètres respectifs de régularisation. Pour obtenir plus de détails au centre du tourbillon avec l'approche locale, on peut réduire le rayon de la fenêtre utilisée, par exemple prendre un rayon de 3 pixels. D'autre part, pour diminuer le bruit présent dans la solution de la méthode globale il faut augmenter le poids relatif à la partie régularisation de l'énergie considérée.
Cependant, les résultats ne sont pas ceux attendus. On voit en FIGURE 2.9 que la solution de l'approche locale est très bruitée alors que celle de la méthode globale, bien que régulière, sous-estime la norme du déplacement dans les zones de vitesse élevée à la périphérie du tourbillon.

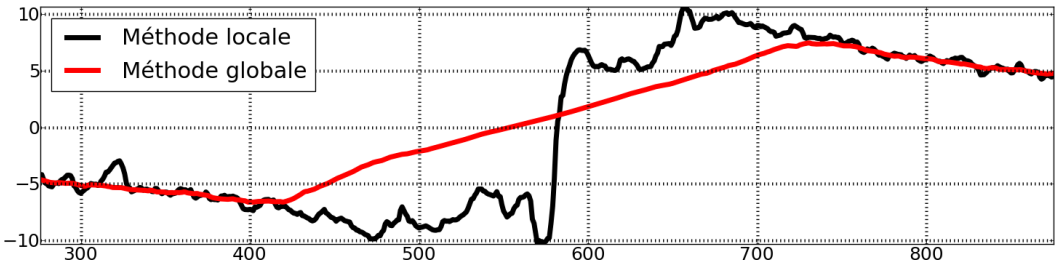

FIGURE 2.9 – Coupe horizontale à travers le cœur du tourbillon (le long de la ligne 333) de la composante verticale des résultats obtenus pour la méthode locale avec un rayon de fenêtre de 3 et la méthode globale avec $\alpha = 100$.

Pour obtenir une solution convenable en termes de précision et de robustesse, jouer sur les paramètres ne suffit pas. Il faut modifier le critère à optimiser. Une méthode idéale devrait être capable de combiner les avantages respectifs des méthodes précitées. Combiner les hypothèses de régularité globale et locale est une piste à envisager. La méthode CLG de Bruhn *et al.* [2005b] a justement été proposée dans cet objectif. La suite de ce chapitre présente plus amplement cette méthode.

2.2 Méthode CLG (*combined local and global*) de Bruhn *et al.*

L'introduction aux méthodes par régularisation locale faite en section 1.2.2.1 indique que le flot optique entre deux images peut être obtenu en résolvant un problème aux moindres carrés. La solution s'obtient par la résolution de l'équation linéaire (1.16) en chaque pixel de l'image :

$$\mathcal{G} * \left(\mathbf{u}(\mathbf{x})\nabla_3 I^T \nabla_3 I \mathbf{u}^T(\mathbf{x})\right). \tag{2.1}$$

D'autre part, l'hypothèse de régularité globale (modélisée dans la partie 1.2.2.2), associée à l'utilisation de la contrainte fondamentale du flot optique (1.4) avec pénalisation quadratique, conduit à la minimisation de la fonctionnelle

$$\iint_\Omega \mathbf{u}\nabla_3 I^T \nabla_3 I \mathbf{u}^T + \alpha\mathcal{R}(\nabla u, \nabla v)dxdy. \tag{2.2}$$

La démarche de Bruhn *et al.* consiste à remplacer le terme d'adéquation aux données de l'énergie (2.2) par le terme (2.1). Ils proposent alors de minimiser le critère CLG (pour "Combined Local Global") :

$$\iint_\Omega \mathcal{G} * \left(\mathbf{u}(\mathbf{x})\nabla_3 I^T \nabla_3 I \mathbf{u}^T(\mathbf{x})\right) + \alpha\mathcal{R}(\nabla u, \nabla v)dxdy. \tag{2.3}$$

2.2.1 Gestion des grands déplacements

Pour correctement estimer les grands déplacements, Bruhn *et al.* mettent en place un schéma itératif "coarse to fine" s'appuyant sur des linéarisations successives de la partie adéquation aux données du critère à minimiser comme décrit dans la partie 1.2.3. À chaque itération, le problème considéré est donc

$$\int_\Omega \mathcal{G} * \Big(\delta\mathbf{u}^n(\mathbf{x})\nabla_3 I^{nT}\nabla_3 I^n \delta\mathbf{u}^{nT}(\mathbf{x})\Big) + \alpha\mathcal{R}(\nabla u^{n+1}, \nabla v^{n+1})dxdy, \tag{2.4}$$

où $I^n(\mathbf{x}) = I(\mathbf{x} + \mathbf{u}^n(\mathbf{x}))$.

Bien que justifié dans le cadre global par Papenberg *et al.* [2006], la méthode de warping ne peut pas s'appliquer directement à l'approche CLG. En effet, la démonstration de Papenberg *et al.* repose sur le développement limité de l'hypothèse de conservation de l'intensité lumineuse (1.30). Ce développement n'est pas valide sous l'hypothèse de régularité locale. Celle-ci impose de considérer le problème de linéarisation de la partie adéquation aux données du point de vue du recalage de fenêtres [Le Besnerais et Champagnat, 2005].

2.3 Justification de l'usage du warping dans CLG

Une méthode par recalage de fenêtre (*i.e.* par régularisation locale) nécessite la comparaison de zones contiguës des images considérées. On cherche alors le vecteur $\mathbf{u}(\mathbf{x})$ qui minimise la somme des écarts quadratiques d'intensité entre le voisinage $\mathcal{V}$ de (x, y) et ce même voisinage translaté par $\mathbf{u}(\mathbf{x})$. Le critère à minimiser devient donc

$$\iint_\Omega \mathcal{G} * \Big[(I(\cdot + \mathbf{u}(\mathbf{x})) - I(\cdot))^2\Big] + \alpha\mathcal{R}(\nabla u, \nabla v)dxdy. \tag{2.5}$$

Le symbole "·" désigne ici la variable muette de l'opération de convolution.

2.3.1 Discussion de l'usage du warping dans CLG de Bruhn *et al.*

En conservant la structure itérative (1.28) et en effectuant un développement de Taylor, la partie adéquation aux données de la fonctionnelle (2.5) devient

$$\begin{aligned}
&\mathcal{G} * \Big[\Big(I(\cdot + \mathbf{u}^{n+1}(\mathbf{x})) - I(\cdot)\Big)^2\Big] \\
&= \iint_{\mathbb{R}^2} \mathcal{G}(x - x', y - y')\big(I(\mathbf{x}' + \mathbf{u}^n(\mathbf{x}) + \delta\mathbf{u}^n(\mathbf{x})) - I(\mathbf{x}')\big)^2 dx'dy' \\
&\approx \iint_{\mathbb{R}^2} \mathcal{G}(x - x', y - y')\Big(\begin{array}{l} I(\mathbf{x}' + \mathbf{u}^n(\mathbf{x})) - I(\mathbf{x}') + \\ \delta u^n(\mathbf{x})\partial_x I(\mathbf{x}' + \mathbf{u}^n(\mathbf{x})) + \\ \delta v^n(\mathbf{x})\partial_y I(\mathbf{x}' + \mathbf{u}^n(\mathbf{x}))\end{array}\Big)^2 dx'dy'
\end{aligned} \tag{2.6}$$

où $\mathbf{x}' = (x', y', t)$.

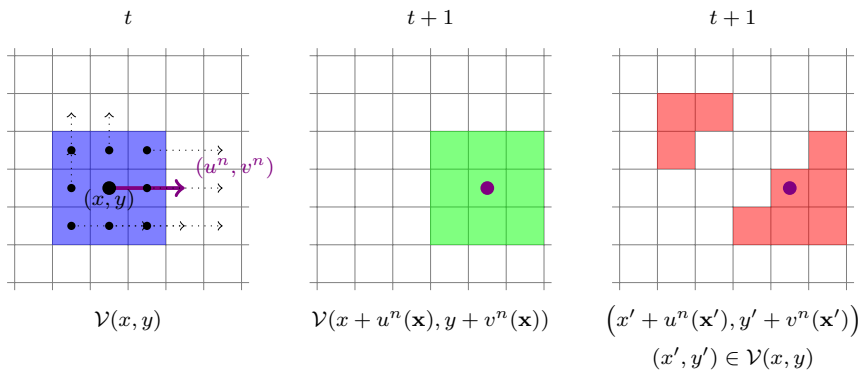

FIGURE 2.10 – Différentes zones du domaine image Ω sur laquelle l'intensité lumineuse et ses dérivées sont évaluées dans les équations (2.6) et (2.8). **À gauche :** zone concernée à l'instant t. **Au centre :** zone à l'instant $t+1$ par un recalage de fenêtres correct (2.6). **À droite :** zone à l'instant $t+1$ considérée lors du warping tel que mis en œuvre par Bruhn *et al.* dans la méthode CLG (2.8).

En général, la formule obtenue (2.6) est différente de celle utilisée par Bruhn *et al.* (2.4) :

$$\mathcal{G} * \Big(\delta\mathbf{u}^n(\mathbf{x})\nabla_3 I^{nT}\nabla_3 I^n \delta\mathbf{u}^{nT}(\mathbf{x})\Big) \tag{2.7}$$

$$= \iint_{\mathbb{R}^2} \mathcal{G}(x-x', y-y')\Big(\begin{array}{l} I(\mathbf{x}' + \mathbf{u}^n(\mathbf{x}')) - I(\mathbf{x}') + \\ \delta u^n(\mathbf{x})\partial_x I(\mathbf{x}' + \mathbf{u}^n(\mathbf{x}')) + \\ \delta v^n(\mathbf{x})\partial_y I(\mathbf{x}' + \mathbf{u}^n(\mathbf{x}'))\Big)^2 dx'dy'. \end{array} \tag{2.8}$$

En effet, l'intensité I et ses dérivées partielles sont évaluées au point $\mathbf{x}' + \mathbf{u}^n(\mathbf{x}')$ alors qu'elles devraient l'être en $\mathbf{x}' + \mathbf{u}^n(\mathbf{x})$ d'après le développement de Taylor (2.6).

Remarque 2.3.1 *L'égalité entre* (2.6) *et* (2.8) *n'intervient que dans deux cas particuliers :*

- $\mathbf{u}^n$ *est uniformément constant.*
- *La gaussienne $\mathcal{G}$ est d'écart type nul, c'est-à-dire qu'on se ramène au cas où $\mathcal{G}$ est une dirac (*i.e. *son support est réduit à un unique pixel). Ce cas correspond à l'application de la régularisation globale seule, traité par Papenberg* et al. *[2006].*

La fonctionnelle (2.4) utilisée par Bruhn *et al.* ne permet pas de minimiser l'énergie non linéaire combinant les approches globale et locale (2.5). Pour illustrer l'approximation effectuée, la FIGURE 2.10 représente les différentes zones de Ω considérées dans les équations (2.6) et (2.8) considérée en un point $\mathbf{x} \in \Omega \times \mathbb{R}$ dans le cas où le flot n'est pas constant sur un voisinage $\mathcal{V}(x, y)$.

Remarque 2.3.2 *La mise en œuvre du warping de Bruhn* et al. *correspond en fait*

au lissage par filtrage gaussien des coefficients de la partie adéquation aux données dans la formulation globale pure (2.2).

Si on veut réellement mettre en œuvre l'approche itérative de l'équation (2.6), on est obligé d'effectuer en tout point (x, y) appartenant à Ω, l'interpolation de l'intensité en $\mathbf{x}' + u(\mathbf{x})$ pour tout point (x', y') appartenant à $\mathcal{V}(x, y)$. Autrement dit, (2.6) implique autant d'interpolation qu'il n'y a de points dans $\mathcal{V}(x, y)$. Cette opération est très coûteuse en temps de calcul.

2.3.2 MCLG ou la mise en œuvre efficace et correcte du warping dans CLG

Pour éviter les calculs d'interpolation superflus induits par la formulation (2.6), l'idée consiste à effectuer le développement limité de l'intensité en $\mathbf{x}' + \mathbf{u}^n(\mathbf{x}')$ plutôt qu'en $\mathbf{x}' + \mathbf{u}^n(\mathbf{x})$:

$$\begin{aligned}
&\mathcal{G} * \left[\Big(I(\cdot + \mathbf{u}^{n+1}(\mathbf{x})) - I(\cdot)\Big)^2\right] \\
&= \iint_{\mathbb{R}^2} \mathcal{G}(x - x', y - y')\Big(I(\mathbf{x}' + \mathbf{u}^n(\mathbf{x}') - \mathbf{u}^n(\mathbf{x}') + \mathbf{u}^{n+1}(\mathbf{x})) - I(\mathbf{x}')\Big)^2 dx'dy' \\
&\approx \iint_{\mathbb{R}^2} \mathcal{G}(x - x', y - y')\Big(\begin{array}{l} I\big(\mathbf{x}' + \mathbf{u}^n(\mathbf{x}')\big) - I(\mathbf{x}') + \\ (u^{n+1}(\mathbf{x}) - u^n(\mathbf{x}'))\partial_x I\big(\mathbf{x}' + \mathbf{u}^n(\mathbf{x}')\big) + \\ (v^{n+1}(\mathbf{x}) - v^n(\mathbf{x}'))\partial_y I\big(\mathbf{x}' + \mathbf{u}^n(\mathbf{x}')\big)\Big)^2 dx'dy'. \end{array}
\end{aligned} \tag{2.9}$$

En adoptant les notations suivantes :

$$\bar{\partial}_t I^n = I^n - I - u^n \partial_x I^n - v^n \partial_y I^n, \tag{2.10}$$

$$\bar{\nabla}_3 I^n = (\partial_x I^n, \partial_y I^n, \bar{\partial}_t I^n), \tag{2.11}$$

l'énergie à minimiser incorporant (2.9) s'écrit

$$\int_\Omega \mathcal{G} * \Big(\mathbf{u}^{n+1}(\mathbf{x})\bar{\nabla}_3 I^{nT}\bar{\nabla}_3 I^n \mathbf{u}^{n+1^T}(\mathbf{x})\Big) + \alpha\mathcal{R}(\nabla u^{n+1}, \nabla v^{n+1})dxdy. \tag{2.12}$$

On retrouve la même structure que dans l'énergie considérée dans le cas de la régularisation globale (2.2) seule ou dans le cas de la méthode CLG de Bruhn *et al.* (2.4). Cette version modifiée de CLG se nommera MCLG pour *modified* CLG.

Remarque 2.3.3 *Ce développement a fait l'objet d'une communication à la conférence européenne du traitement du signal [Fezzani* et al.*, 2010].*

2.4 Évaluation ampirique de la combinaison des méthodes locale et globale

On est en droit de se demander à ce niveau, si cette formulation atteint le but fixé à la suite des expériences menées au début de ce chapitre. Les FIGURES 2.11 et

2.12 reproduisent les résultats de la méthode MCLG (2.12) avec une régularisation quadratique.
L'examen qualitatif de ces résultats montre le succès de cette méthode pour offrir le compromis entre la précision de la régularisation globale et la robustesse de l'approche locale.
On observe tout d'abord la précision de la méthode MCLG au niveau des zones du cœur des tourbillons. La solution présente en effet des courbes circulaires lisses et concentriques.
D'autre part, le bruit caractéristique observé dans les solutions de la méthode globale au niveau des zones de faible déplacement est nettement réduit dans la solution de la méthode MCLG.

Ces résultats préliminaires seront complétés dans le chapitre 6.

2.5 Conclusion

Au court de ce chapitre, l'étude ampirique des méthodes locale et globale est effectuée dans le cadre particulier de l'application des techniques d'estimation du flot optique à la vélocimétrie par images de particules. Nous montrons ainsi l'interrés que peut représenter la combinaison de ces deux approches, ce qui nous mène à étudier la méthode CLG introduite par Bruhn *et al.* [2005b].
Au court de cette étude, nous exhibons une approximation faite par Bruhn *et al.* [2005b] dans la mise en œuvre du warping dans CLG. Nous proposons donc une correction efficace de ce schéma et montrons le compromis offert par la combinaison des approches locale et globale en PIV.

Nous décrivons maintenant différents algorithmes de minimisation du critère MCLG dans les chapitres 3 et 4.

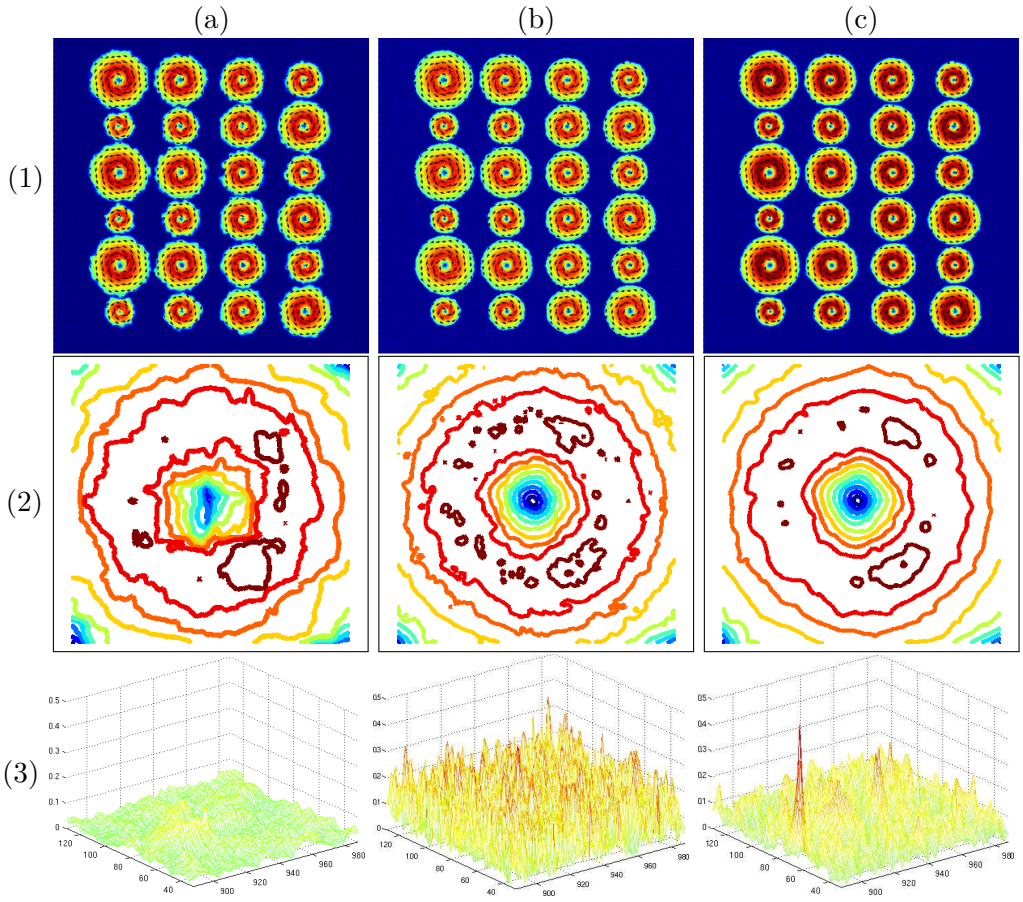

FIGURE 2.11 – Traitement du cas synthétique par la méthode MCLG avec un rayon de fenêtre de 3 et $\alpha = 60$ comparé avec les résultats des méthodes locale et globale présenté dans les FIGURES 2.2, 2.3 et 2.4. **(1) :** norme du résultat. **(2) :** Contour des normes sur la zone du carré rouge (*c.f.*FIGURE 2.1). **(3) :** Norme en relief de la zone du carré vert (*c.f.*FIGURE 2.1). **(a) :** résultats de la méthode locale. **(b) :** résultats de la méthode globale. **(c) :** résultats de la méthode MCLG. Le résultat de MCLG présente un compromis entre le respect des zones de fort gradient (image de contour deuxième ligne) et le lissage du bruit des zones de faible déplacement (surface 3D troisième ligne).

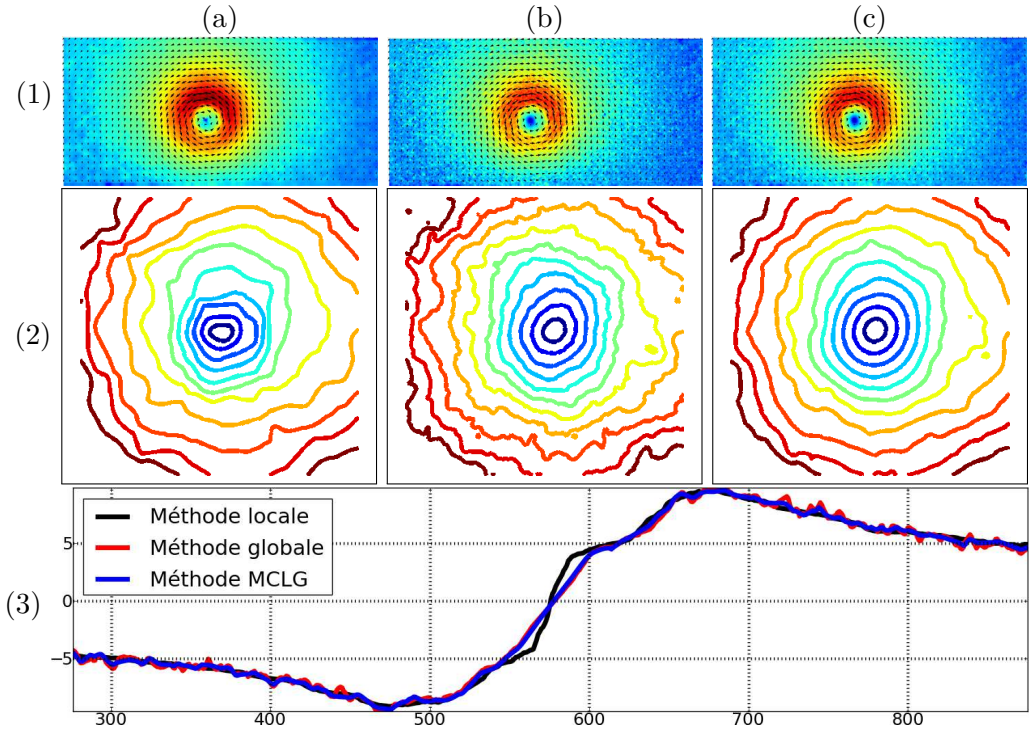

FIGURE 2.12 – Traitement du cas réel par la méthode MCLG avec un rayon de fenêtre de 3 et $\alpha = 60$, comparé avec les FIGURES 2.6, 2.7 et 2.8. **(1) :** norme du résultat. **(2) :** Contour des normes au niveau du cœur du tourbillon. **(3) :** Coupe horizontale à travers le cœur du tourbillon (le long de la ligne 333) de la composante verticale. **(a) :** résultats de la méthode locale. **(b) :** résultats de la méthode globale. **(c) :** résultats de la méthode MCLG. On voit la précision de la méthode MCLG au niveau du centre du tourbillon (*c.f.* (2)) ainsi que le lissage du bruit présents dans les zones de faible déplacement dans la solution de la méthode globale (*c.f.* (3) les oscillation de la méthode globale en s'éloignant du centre du tourbillon).

Chapitre 3

Calcul variationnel

Sommaire

3.1 Équations d'Euler-Lagrange **33**
3.2 Discrétisation de la partie régularisation **35**
3.2.1 Approche classique 35
3.2.2 Approche alternative 35
3.3 Élaboration du solveur non linéaire **37**
3.3.1 Linéarisation locale des équations 38
3.3.2 Méthode SOR non linéaire par bloc 2×2 39
3.4 Parallélisation du solveur **40**
3.4.1 Coloration de mailles "*Red-Black*" 41
3.4.2 Coloration de mailles à quatre couleurs 41
3.5 Conclusion **42**

Ce chapitre s'intéresse à la minimisation du critère MCLG introduit au chapitre 2 par approche variationnelle. Après la mise en place d'un schéma de *warping* modélisant correctement la condition de régularité locale dans le chapitre précédent, cette partie s'intéresse à la minimisation par méthode variationnelle de la fonctionnelle considérée. Le choix de la partie régularisation $\mathcal{R}(\nabla u^{n+1}, \nabla v^{n+1})$ de la fonctionnelle (2.12) est discuté et explicité. À la suite de cela, nous écrivons les équations d'Euler-Lagrange associées à l'énergie considérée. S'ensuit une étude des différentes discrétisations possibles. L'approche classique est décrite puis une méthode alternative est développée.
Ensuite, la non-linarité des équations discrètes obtenues est traitées par la mise en œuvre d'un solveur de type Picard-SOR n'employant que des linéarisations locales du système. De plus, le problème étant posé pour deux inconnues en chaque pixel (composantes horizontale et verticale du flot optique), la résolution doit se faire par blocs 2×2.
Pour finir, la décorrélation des opérations à l'aide d'une méthode de coloration de mailles permet la parallélisation du solveur et ainsi l'accélération de la résolution sur des architectures adaptées.

3.1 Équations d'Euler-Lagrange

Pour l'élaboration d'une méthode de flot optique permettant le calcul de champ de vecteurs discontinus, deux types de régularisation peuvent être utilisés :

– régularisation dépendant des images traitées tel que l'approche de Alvarez *et al.* [1999] (1.21) et celle de Nagel et Enkelmann [1986] (1.22),
– emploi d'une fonction pénalisante robuste sur le terme de régularisation(1.24) comme Cohen [1993] ou encore Blanc-Feraud *et al.* [1993].

Pour le cas particulier de la vélocimétrie par images de particules, l'usage d'une régularisation dépendant des gradients d'images semble inapproprié à cause de l'absence de bords d'objets pouvant désigner une discontinuité du champ de vecteurs (voir les FIGURES 1.5, 2.1 et 2.5 pour s'en convaincre).

Comme par exemple Bruhn *et al.* [2005b], Papenberg *et al.* [2006] où encore Zach *et al.* [2007], nous choisissons d'effectuer la régularisation du flot optique au sens de la variation totale. Une fonction de pénalisation non quadratique est également appliquée au terme d'adéquation aux données afin d'assurer la robustesse de la méthode aux zones d'occultation [Black et Anandan, 1996].
La fonctionnelle que nous considérons dans ces travaux est donc

$$\iint_{\Omega} |\mathcal{G} * \left(\mathbf{u}^{n+1}(\mathbf{x})\bar{\nabla}_3 I^{n^T}\bar{\nabla}_3 I^n \mathbf{u}^{n+1^T}(\mathbf{x})\right)| + \alpha(|\nabla u(\mathbf{x})| + |\nabla v(\mathbf{x})|)dxdy. \tag{3.1}$$

Le fait que la fonction $|.|$ soit non différentiable en zéro interdit l'utilisation des méthodes variationnelles avec la régularisation choisie ici. Une méthode consiste à approcher la variation totale à l'aide de fonctions continûment dérivables. La fonction $\psi(x^2) = \sqrt{x^2 + \varepsilon}$ permet cela.

La partie adéquation aux données de l'énergie (3.1) devient alors

$$\psi\left(\mathcal{G} * \left(\mathbf{u}^{n+1}(\mathbf{x})\bar{\nabla}_3 I^{n^T}\bar{\nabla}_3 I^n \mathbf{u}^{n+1^T}(\mathbf{x})\right)\right). \tag{3.2}$$

Pour la partie régularisation , ψ peut être appliquée de deux manières :
régularisation isotrope [Papenberg *et al.*, 2006; Bruhn *et al.*, 2005b]

$$|\nabla u| + |\nabla v| \approx \psi(|\nabla u|^2 + |\nabla v|^2). \tag{3.3}$$

Les équations d'Euler-Lagrange associées sont alors

$$\Psi_D \cdot \mathcal{G} * \left(\partial_x I^n \left(\bar{\nabla}_3 I^n \mathbf{u}^{n+1}(\mathbf{x})\right)\right) = \alpha \mathrm{div}(\psi'(|\nabla u^{n+1}|^2 + |\nabla v^{n+1}|^2)\nabla u^{n+1}) \tag{3.4}$$

$$\Psi_D \cdot \mathcal{G} * \left(\partial_y I^n \left(\bar{\nabla}_3 I^n \mathbf{u}^{n+1}(\mathbf{x})\right)\right) = \alpha \mathrm{div}(\psi'(|\nabla u^{n+1}|^2 + |\nabla v^{n+1}|^2)\nabla v^{n+1}) \tag{3.5}$$

régularisation anisotrope [Weickert et Schnorr, 2001]

$$|\nabla u| + |\nabla v| \approx \psi(|\nabla u|^2) + \psi(|\nabla v|^2). \tag{3.6}$$

Les équations d'Euler-Lagrange correspondantes sont

$$\Psi_D \cdot \mathcal{G} * \left(\partial_x I^n \left(\bar{\nabla}_3 I^n \mathbf{u}^{n+1}(\mathbf{x})\right)\right) = \alpha \mathrm{div}(\psi'(|\nabla u^{n+1}|^2)\nabla u^{n+1}) \tag{3.7}$$

$$\Psi_D \cdot \mathcal{G} * \left(\partial_y I^n \left(\bar{\nabla}_3 I^n \mathbf{u}^{n+1}(\mathbf{x})\right)\right) = \alpha \mathrm{div}(\psi'(|\nabla v^{n+1}|^2)\nabla v^{n+1}). \tag{3.8}$$

Ici $\Psi_D = \psi' \left(\mathcal{G} * \left(\mathbf{u}^{n+1}(\mathbf{x}) \bar{\nabla}_3 I^{nT} \bar{\nabla}_3 I^n \mathbf{u}^{n+1^T}(\mathbf{x}) \right) \right)$.

Disposant de données mesurées sur une grille cartésienne uniforme, la méthode des différences finies est un outil de choix, elle est pour cela le moyen de discrétisation le plus utilisé en traitement d'images. La prochaine partie présente différentes discrétisations possibles pour les termes de régularisation isotrope (3.3) et anisotrope (3.6).

3.2 Discrétisation de la partie régularisation

u^{n+1} et v^{n+1} jouent un rôle symétrique dans (3.4)-(3.5) et (3.7)-(3.8). La partie régularisation de ces équations est alors développée pour $(w, \tilde{w}) \in \{(u^{n+1}, v^{n+1}), (v^{n+1}, u^{n+1})\}$.

D'autre part, les données et les inconnues du problème étant considérées sur une grille cartésienne homogène, on considère que le pas de discrétisation est égal à 1 et on choisit d'utiliser la notation matricielle pour indiquer la position $(x, y) \in \Omega$ où est évaluée une expression * :

$$* \, (x, y) = *_{i,j}. \tag{3.9}$$

3.2.1 Approche classique

Le schéma classique décrit par Bruhn *et al.* [2006] est obtenu en combinant l'approximation des opérateurs ∂_x et ∂_y par les différences finies à droite et à gauche. Le détail de ces opérations est présenté dans l'annexe A. Le liens avec la minimisation de la forme discrète du problème y est également décrit.

En notant

- $\Psi_{Ri,j} = \psi'(|\nabla w_{i,j}|^2)$ dans le cas anisotrope (3.7)-(3.8),
- $\Psi_{Ri,j} = \psi'(|\nabla w_{i,j}|^2 + |\nabla \tilde{w}_{i,j}|^2)$ dans le cas isotrope (3.4)-(3.5),

la partie régularisation des équations d'Euler-Lagrange s'écrit en tout point de Ω

$$\operatorname{div}(\Psi_{Ri,j} \nabla w_{i,j}) = \partial_x(\Psi_{Ri,j} \partial_x w_{i,j}) + \partial_y(\Psi_{Ri,j} \partial_y w_{i,j}) \tag{3.10}$$

$$\begin{aligned} \approx \frac{1}{2} \Big(& (\Psi_{Ri,j} + \Psi_{Ri,j+1}) \cdot (w_{i,j+1} - w_{i,j}) - \\ & (\Psi_{Ri,j-1} + \Psi_{Ri,j}) \cdot (w_{i,j} - w_{i,j-1}) + \\ & (\Psi_{Ri,j} + \Psi_{Ri+1,j}) \cdot (w_{i+1,j} - w_{i,j}) - \\ & (\Psi_{Ri,j} + \Psi_{Ri-1,j}) \cdot (w_{i,j} - w_{i-1,j}) \Big) . \end{aligned} \tag{3.11}$$

On notera que ce schéma de discrétisation ne fait intervenir que les quatre voisins du point (i, j) auquel il est évalué.

3.2.2 Approche alternative

L'idée proposée ici est de développer les termes de régularisation. L'utilisation de la relation entre les dérivées première et seconde de ψ permet la factorisation et

la simplification des expressions obtenues.

3.2.2.1 Régularisation anisotrope

En conservant les notations introduites au début de cette section, la partie régularisation des équations d'Euler-Lagrange (3.7)-(3.8) s'écrit

$$\begin{aligned}
\operatorname{div}(\psi'(|\nabla w|^2)\nabla w) &= \partial_x(\psi'(|\nabla w|^2)\cdot\partial_x w) + \partial_y(\psi'(|\nabla w|^2)\cdot\partial_y w) && (3.12)\\
&= \partial^2_{xx} w\cdot\psi'(|\nabla w|^2) + \partial_x w\cdot\partial_x\psi'(|\nabla w|^2) + \\
&\quad \partial^2_{yy} w\cdot\psi'(|\nabla w|^2) + \partial_y w\cdot\partial_y\psi'(|\nabla w|^2). && (3.13)
\end{aligned}$$

D'autre part,

$$\begin{aligned}
\partial_x\psi'(|\nabla w|^2) &= 2(\partial^2_{xx} w\cdot\partial_x w + \partial^2_{xy} w\cdot\partial_y w)\cdot\psi''(|\nabla w|^2) && (3.14)\\
\partial_y\psi'(|\nabla w|^2) &= 2(\partial^2_{yy} w\cdot\partial_y w + \partial^2_{xy} w\cdot\partial_x w)\cdot\psi''(|\nabla w|^2). && (3.15)
\end{aligned}$$

En remarquant que $\psi''(x) = \dfrac{-\psi'(x)}{2(x+\varepsilon)}$, on obtient :

$$\begin{aligned}
\operatorname{div}(\psi'(|\nabla w|^2)\nabla w) = {} & \psi'(|\nabla w|^2)(\partial^2_{xx} w + \partial^2_{yy} w) - \\
& \frac{\psi'(|\nabla w|^2)}{|\nabla w|^2+\varepsilon}\Big(\begin{array}{l}\partial^2_{xx} w\cdot(\partial_x w)^2 + \\ \partial^2_{yy} w\cdot(\partial_y w)^2 + \\ 2\partial^2_{xy} w\cdot\partial_x w\partial_y w\Big)\end{array} && (3.16)\\
= {} & \frac{\psi'(|\nabla w|^2)}{|\nabla w|^2+\varepsilon}\Big(\begin{array}{l}((\partial_y w)^2+\varepsilon)\partial^2_{xx} w + \\ ((\partial_x w)^2+\varepsilon)\partial^2_{yy} w - \\ (2\partial_x w\cdot\partial_y w)\partial^2_{xy} w\Big).\end{array} && (3.17)
\end{aligned}$$

La discrétisation de cette équation se fait en utilisant le schéma centré pour les dérivées partielles du premier ordre et les schémas classiques suivants pour celles du second ordre :

$$\begin{aligned}
\partial^2_{xx} w_{i,j} &\approx w_{i,j+1} + w_{i,j-1} - 2w_{i,j}\ , && (3.18)\\
\partial^2_{yy} w_{i,j} &\approx w_{i+1,j} + w_{i-1,j} - 2w_{i,j}\ , && (3.19)\\
\partial^2_{xy} w_{i,j} &\approx \frac{1}{4}(w_{i-1,j-1} - w_{i+1,j-1} + w_{i+1,j+1} - w_{i-1,j+1}). && (3.20)
\end{aligned}$$

On remarquera cette fois-ci l'utilisation des huit voisins du point de Ω auquel est considérée l'équation.

3.2.2.2 Régularisation isotrope

La même démarche que dans le cas de la régularisation anisotrope est appliquée ici aussi. La partie régularisation des équations d'Euler-Lagrange (3.4)-(3.5) s'écrit

$$\begin{aligned}
\operatorname{div}(\psi'(|\nabla w|^2+|\nabla\tilde{w}|^2)\nabla w) = {} & \partial_x(\psi'(|\nabla w|^2+|\nabla\tilde{w}|^2)\cdot\partial_x w)+ \\
& \partial_y(\psi'(|\nabla w|^2+|\nabla\tilde{w}|^2)\cdot\partial_y w) && (3.21)\\
= {} & \partial^2_{xx}w\cdot\psi'(|\nabla w|^2+|\nabla\tilde{w}|^2)+ \\
& \partial_x w\cdot\partial_x\psi'(|\nabla w|^2+|\nabla\tilde{w}|^2)+ \\
& \partial^2_{yy}w\cdot\psi'(|\nabla w|^2+|\nabla\tilde{w}|^2)+ \\
& \partial_y w\cdot\partial_y\psi'(|\nabla w|^2+|\nabla\tilde{w}|^2). && (3.22)
\end{aligned}$$

Or les dérivées partielles de $\psi'(|\nabla w|^2+|\nabla\tilde{w}|^2)$ sont

$$\begin{aligned}
\partial_x\psi'(|\nabla w|^2+|\nabla\tilde{w}|^2) = 2\big(& \partial^2_{xx}w\cdot\partial_x w+\partial^2_{xy}w\cdot\partial_y w+ \\
& \partial^2_{xx}\tilde{w}\cdot\partial_x\tilde{w}+\partial^2_{xy}\tilde{w}\cdot\partial_y\tilde{w}\big)\cdot\psi''(|\nabla w|^2+|\nabla\tilde{w}|^2) && (3.23)\\
\partial_y\psi'(|\nabla w|^2+|\nabla\tilde{w}|^2) = 2\big(& \partial^2_{yy}w\cdot\partial_y w+\partial^2_{xy}w\cdot\partial_x w+ \\
& \partial^2_{yy}\tilde{w}\cdot\partial_y\tilde{w}+\partial^2_{xy}\tilde{w}\cdot\partial_x\tilde{w}\big)\cdot\psi''(|\nabla w|^2+|\nabla\tilde{w}|^2), && (3.24)
\end{aligned}$$

et finalement, le terme divergence (3.22) devient

$$\frac{\psi'(|\nabla w|^2+|\nabla\tilde{w}|^2)}{|\nabla w|^2+|\nabla\tilde{w}|^2+\varepsilon}\Big(\begin{array}{ll}
(\partial_y w^2+\partial_x\tilde{w}^2+\partial_y\tilde{w}^2+\varepsilon) & \partial^2_{xx}w+ \\
(\partial_x w^2+\partial_x\tilde{w}^2+\partial_y\tilde{w}^2+\varepsilon) & \partial^2_{yy}w- \\
2\partial_x w\partial_y w & \partial^2_{xy}w- \\
\partial_x w\partial_x\tilde{w} & \partial^2_{xx}\tilde{w}- \\
\partial_y w\partial_y\tilde{w} & \partial^2_{yy}\tilde{w}- \\
(\partial_x w\partial_y\tilde{w}+\partial_y w\partial_x\tilde{w}) & \partial^2_{xy}\tilde{w}
\end{array}\Big). \qquad (3.25)$$

La discrétisation de ces équations se fait comme pour le cas de la régularisation anisotrope (3.17) en utilisant les différences centrées pour les dérivées partielles du premier ordre et les schémas (3.18), (3.19) et (3.20) pour celles à l'ordre deux. Cette discrétisation fait également intervenir les huit plus proches voisins du point (i,j) considéré.

Remarque 3.2.1 *Le développement de la régularisation isotrope* (3.25)*, en faisant intervenir les valeurs de w et $\tilde{w}$, entraîne la résolution d'un système avec seize interactions.*

3.3 Élaboration du solveur non linéaire

Les équations d'Euler-Lagrange (3.4)-(3.5) et (3.7)-(3.8) discrétisées à l'aide des différents schémas (3.11), (3.17) et (3.25) présentés à la section précédente donnent lieu à un ensemble de systèmes d'équations fortement non linéaires.
Pour leur résolution, nous emploierons uniquement des linéarisations locales ne faisant intervenir que les voisins du point considéré. Les méthodes de relaxation sont dans ce cadre un outil idéal.

3.3.1 Linéarisation locale des équations

En tout point (i, j) de la grille, les différents systèmes produits dans ce chapitres par la discrétisation des équations peuvent s'écrire sous la forme

$$\begin{aligned}\mathcal{N}^u_{i,j}(\ & u^{n+1}_{i-1,j-1}, v^{n+1}_{i-1,j-1}, u^{n+1}_{i-1,j}, v^{n+1}_{i-1,j}, u^{n+1}_{i-1,j+1}, v^{n+1}_{i-1,j+1}, \\ & u^{n+1}_{i,j-1}, v^{n+1}_{i,j-1}, u^{n+1}_{i,j}, v^{n+1}_{i,j}, u^{n+1}_{i,j+1}, v^{n+1}_{i,j+1}, \\ & u^{n+1}_{i+1,j-1}, v^{n+1}_{i+1,j-1}, u^{n+1}_{i+1,j}, v^{n+1}_{i+1,j}, u^{n+1}_{i+1,j+1}, v^{n+1}_{i+1,j+1}\ ) = 0,\end{aligned} \tag{3.26}$$

$$\begin{aligned}\mathcal{N}^v_{i,j}(\ & u^{n+1}_{i-1,j-1}, v^{n+1}_{i-1,j-1}, u^{n+1}_{i-1,j}, v^{n+1}_{i-1,j}, u^{n+1}_{i-1,j+1}, v^{n+1}_{i-1,j+1}, \\ & u^{n+1}_{i,j-1}, v^{n+1}_{i,j-1}, u^{n+1}_{i,j}, v^{n+1}_{i,j}, u^{n+1}_{i,j+1}, v^{n+1}_{i,j+1}, \\ & u^{n+1}_{i+1,j-1}, v^{n+1}_{i+1,j-1}, u^{n+1}_{i+1,j}, v^{n+1}_{i+1,j}, u^{n+1}_{i+1,j+1}, v^{n+1}_{i+1,j+1}\ ) = 0.\end{aligned} \tag{3.27}$$

On utilise la méthode de Picard [Trottenberg *et al.*, 2001] pour linéariser ces équations ,*i.e.*, en se plaçant dans un cadre itératif, certains coefficients sont fixés à l'aide des itérées précédents :

- pour la méthode classique de discrétisation (3.11), ce sont les coefficients Ψ_R qui sont fixés [Vogel et Oman, 1996]. Cette méthode est connue en traitement d'images sous le nom de "*lagged diffusivity fixed point*" [Chan et Mulet, 1999],
- en ce qui concerne l'approche proposée dans cette thèse, les dérivées partielles du premier ordre qui apparaissent dans les développements (3.17) et (3.25) sont fixées.

Remarque 3.3.1 *Les notations* (3.26) *et* (3.27) *introduites ici représentent le cas général où les valeurs de* u^{n+1} *et* v^{n+1} *sont évaluées aux huit voisins du point* (i, j)*. Les coefficients des systèmes localement linéarisés seront détaillés dans l'annexe B pour chacune des équations présentées dans ce chapitre.*

La forte non-linéarité des équations manipulées implique le besoins de re-linéariser de nombreuses fois le système par la méthode de Picard. Nous choisissons donc d'utiliser la méthode de Gauss-Seidel, non linéaire, avec sur-relaxations successives (SOR, décrite dans le cadre linéaire par Saad [2003] ou encore Golub et Van Loan [1996]) pour la résolution des équations localement linéarisées à la volée. Ainsi, en utilisant cette procédure, aucun assemblage de matrice n'est nécessaire. Ceci nous permet de plus d'économiser de l'espace mémoire.

Par ailleurs, nous avons testé l'approche de linéarisation globale au début de la thèse [Le Besnerais *et al.*, 2009] dans l'esprit de Bruhn *et al.* [2006, 2005a]. Cette approche a ensuite été abandonnée car elle s'est avérée peu adaptée pour correctement traiter le caractère non-linéaire des équations considérées.

Remarque 3.3.2 *Les performances "temps réel" affichés par Bruhn* et al. *[2006, 2005a] doivent être relativisées au vu de la dimension très réduite (*200×200*) des images qu'ils utilisent dans leurs expériences.*

3.3.2 Méthode SOR non linéaire par bloc 2×2

L'introduction d'une nouvelle boucle itérative implique l'ajout d'une variable d'itération $k \in \mathbb{N}$ à la notation des inconnues du problème :

$$\mathbf{u}^{n,k} = (u^{n,k}, v^{n,k}, 1) \tag{3.28}$$

où n est l'indice relatif à la boucle de warping et k à la méthode de relaxation.

La méthode de Gauss-Seidel appliquée au problème consiste à parcourir les indices i et j et à déduire $\mathbf{u}_{i,j}^{n+1,k+1}$ à partir des systèmes locaux $\mathcal{N}_{i,j}^u$ et $\mathcal{N}_{i,j}^v$ qui lui sont associés en fixant les autres inconnues de l'équation à l'aide d'éléments déjà calculés.

L'application naïve de ce principe revient à résoudre

$$\mathcal{N}_{i,j}^u\left(\begin{array}{lllllll} u_{i-1,j-1}^{n+1,k+1} & , v_{i-1,j-1}^{n+1,k+1} & , u_{i-1,j}^{n+1,k+1} & , v_{i-1,j}^{n+1,k+1} & , u_{i-1,j+1}^{n+1,k+1} & , v_{i-1,j+1}^{n+1,k+1} & , \\ u_{i,j-1}^{n+1,k+1} & , v_{i,j-1}^{n+1,k+1} & , u_{i,j}^{n+1,k+1} & , v_{i,j}^{n+1,k} & , u_{i,j+1}^{n+1,k} & , v_{i,j+1}^{n+1,k} & , \\ u_{i+1,j-1}^{n+1,k} & , v_{i+1,j-1}^{n+1,k} & , u_{i+1,j}^{n+1,k} & , v_{i+1,j}^{n+1,k} & , u_{i+1,j+1}^{n+1,k} & , v_{i+1,j+1}^{n+1,k} & \end{array}\right) = 0, \tag{3.29}$$

pour obtenir $u_{i,j}^{n+1,k+1}$, suivi de la résolution de

$$\mathcal{N}_{i,j}^v\left(\begin{array}{lllllll} u_{i-1,j-1}^{n+1,k+1} & , v_{i-1,j-1}^{n+1,k+1} & , u_{i-1,j}^{n+1,k+1} & , v_{i-1,j}^{n+1,k+1} & , u_{i-1,j+1}^{n+1,k+1} & , v_{i-1,j+1}^{n+1,k+1} & , \\ u_{i,j-1}^{n+1,k+1} & , v_{i,j-1}^{n+1,k+1} & , u_{i,j}^{n+1,k+1} & , v_{i,j}^{n+1,k+1} & , u_{i,j+1}^{n+1,k} & , v_{i,j+1}^{n+1,k} & , \\ u_{i+1,j-1}^{n+1,k} & , v_{i+1,j-1}^{n+1,k} & , u_{i+1,j}^{n+1,k} & , v_{i+1,j}^{n+1,k} & , u_{i+1,j+1}^{n+1,k} & , v_{i+1,j+1}^{n+1,k} & \end{array}\right) = 0. \tag{3.30}$$

pour le calcul de $v_{i,j}^{n+1,k+1}$.

Cette démarche séquentielle a le défaut de toujours mettre à jour $v_{i,j}^{n+1}$ après $u_{i,j}^{n+1}$. Une approche idéale permettrait le calcul des deux composantes du flot en une seule étape, d'où l'idée de la résolution par blocs.

Pour formaliser la résolution du système d'équations à l'aide de la méthode de Gauss-Seidel par blocs, on ramène les étapes (3.29) et (3.30) sous la forme

$$A \cdot u_{i,j}^{n+1,k+1} + B \cdot v_{i,j}^{n+1,k} + D = 0 \tag{3.31}$$

$$E \cdot u_{i,j}^{n+1,k+1} + F \cdot v_{i,j}^{n+1,k+1} + G = 0 \tag{3.32}$$

La résolution par blocs revient alors à l'inversion du système 2×2

$$\begin{pmatrix} A & B \\ E & F \end{pmatrix} \begin{pmatrix} u_{i,j}^{n+1} \\ v_{i,j}^{n+1} \end{pmatrix}^{k+1} + \begin{pmatrix} D \\ G \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{3.33}$$

De cette manière, $\begin{pmatrix} u_{i,j}^{n+1} \\ v_{i,j}^{n+1} \end{pmatrix}^{k}$ est mis à jour par une même itération du solveur.

La convergence de la méthode de Gauss-Seidel peut être accélérée en la surrelaxant. C'est ce que propose la méthode SOR (*Successive Over Relaxation*).

En définissant $1 \leq \omega \leq 2$, et en notant

$$\begin{pmatrix} u_{i,j}^{n+1} \\ v_{i,j}^{n+1} \end{pmatrix}_{\mathcal{GS}}^{k+1} \tag{3.34}$$

le résultat de la mise à jour de $\begin{pmatrix} u_{i,j}^{n+1} \\ v_{i,j}^{n+1} \end{pmatrix}^{k}$ par la méthode de Gauss-Seidel (3.33), une itération de SOR s'écrit

$$\begin{pmatrix} u_{i,j}^{n+1} \\ v_{i,j}^{n+1} \end{pmatrix}^{k+1} = \omega \begin{pmatrix} u_{i,j}^{n+1} \\ v_{i,j}^{n+1} \end{pmatrix}_{\mathcal{GS}}^{k+1} + (1-\omega) \begin{pmatrix} u_{i,j}^{n+1} \\ v_{i,j}^{n+1} \end{pmatrix}^{k}. \tag{3.35}$$

L'approche proposée peut être relativement coûteuse en temps de calcul. En effet, une paire d'images de dimension $nl \times nc$ engendre un système de $2 \times nl \times nc$ inconnues. Le nombre d'itérations de SOR nécessaires à la convergence peut être élevé. Une manière d'accélérer le traitement est de décorréler les opérations afin de permettre la parallélisation de l'algorithme et de pouvoir profiter de la puissance de calcul qu'offrent les architectures multiprocesseurs actuelles.

Remarque 3.3.3 *Pour accélérer la convergence des méthodes de relaxation, une approche multigrille peut être envisagée. Au cours de ces travaux, la résolution par multigrille linéaire [Wesseling, 1991] s'appuyant sur une linéarisation globale du système a été écartée car peu adaptée au traîtement du caractère non-linéaire des équations. L'élaboration d'un solveur multigrille non-linéaire (de type FAS [Trottenberg* et al.*, 2001] ou encore MNM [Yavneh et Dardyk, 2006]) a alors été entrepris, mais cette voie n'a pas abouti.*

3.4 Parallélisation du solveur

L'étude bibliographique de la section 1.3.1 montre l'efficacité de la parallélisation à grain fin des méthodes de flot optique.

De plus il est possible d'inclure de tels algorithmes dans un schéma de parallélisation à mémoire distribuée comme le stipule la remarque 1.3.1. L'annexe C explique cette possibilité appliquée au solveur développé ici (qu'il soit parallèle ou non).

On s'intéresse donc dans cette partie à la parallélisation à grain fin du solveur SOR développé à la section précédente.

La structure particulière des dépendances entre les mailles de la grille cartésienne uniforme sur laquelle est calculée la solution du problème du flot optique offre la possibilité d'employer des méthodes de coloriage de mailles pour la décorrélation des opérations au cours des itérations de SOR [Saad, 2003; Adams et Ortega, 1982]. Cette méthode consiste à "colorer" les mailles de la grille sur laquelle est résolue le

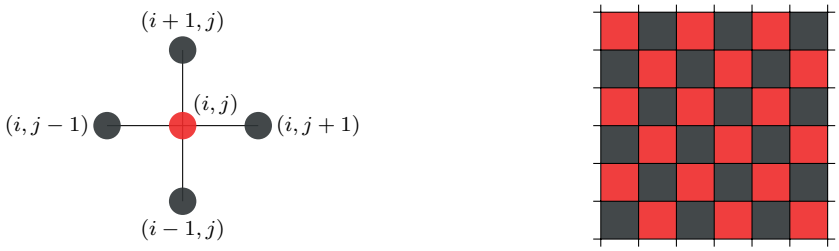

FIGURE 3.1 – Coloriage des mailles avec deux couleurs. **À gauche :** pattern des interactions entre voisins. **À droite :** Grille coloriée.

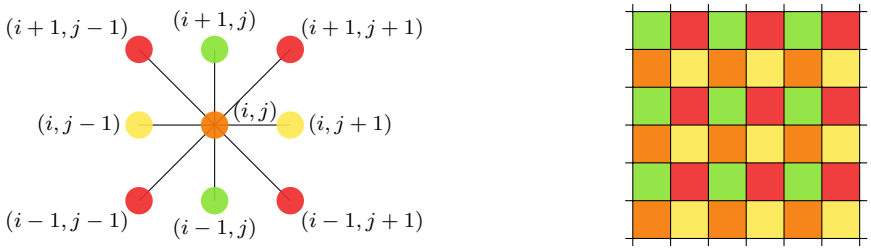

FIGURE 3.2 – Coloriage des mailles avec quatre couleurs. **À gauche :** pattern des interactions entre voisins. **À droite :** Grille coloriée.

problème de sorte que tous les voisins intervenant dans le calcul d'un point soient d'une couleur différente de ce dernier.
Les points appartenant à une même couleur peuvent ainsi être mis à jour indépendamment les uns des autres. La stratégie revient donc à regrouper les points par couleur et à résoudre les équations, un groupe après l'autre, parallèlement.

3.4.1 Coloration de mailles "*Red-Black*"

La discrétisation classique (3.11) des équations d'Euler-Lagrange avec régularisation isotrope (3.4)-(3.5) ou anisotrope (3.7)-(3.8) produit un schéma avec quatre interactions par point de grille.
La FIGURE 3.1 représente le pattern des interactions et le maillage tel qu'il est coloré. La coloration à l'aide de deux couleurs est dite méthode "*red-black*".

3.4.2 Coloration de mailles à quatre couleurs

Les approches alternatives (3.17) et (3.25) proposées pour la discrétisation des équations d'Euler-Lagrange engendrent pour leurs part un schéma avec une interaction avec l'ensemble des huit plus proches voisins en chaque point de grille.
Quatre couleurs sont nécessaires dans ce cas pour isoler chaque maille et permettre la décorrélation de la mise à jour des points de même couleur.
La FIGURE 3.2 montre les interactions entre un point et ses voisins ainsi que la manière de colorer le maillage.

Remarque 3.4.1 *Dans le cadre de la linéarisation locale par la méthode de Picard, le choix du schéma centré pour fixer les dérivées partielles du premier ordre permet, après coloriage du maillage, de garantir que les points utilisés pour calculer ces dérivées sont au même niveau d'itération.*

3.5 Conclusion

À l'issue de ce chapitre, un solveur paralléle par blocs 2×2 de type SOR non linéaire a été élaboré. Le modèle choisit pour la régularisation est celui de la pénalisation de la variation totale. Nous avons par ailleurs présenté une nouvelle méthode pour la discrétisation des équations d'Euler-Lagrange de ce critère.

Malgré l'accélération notable qu'offre la résolution parallèle par rapport à un traitement séquentiel du problème, le temps de calcul des méthodes par calcul variationnel peut être de plusieurs secondes pour des cas réels d'applications où les images considérées sont de grande taille.

Le prochain chapitre est consacré à la description d'une technique de "*variable splitting*" basée sur la méthode de Bregman et à son application à l'optimisation du modèle MCLG (2.12). Cette approche a l'avantage d'impliquer uniquement des opérations décorrélées, sa parallélisation est de fait plus aisée.

Chapitre 4

Résolution par la méthode de (“*variable splitting*”)

Sommaire

4.1 Reformulation du problème de calcul du flot optique 43
4.2 Méthode de Bregman 45
4.3 Application à la méthode MCLG 47
4.3.1 Résolution de la partie adéquation aux données 47
4.3.2 Résolution de la partie régularisation 48
4.3.3 Parallélisation 49
4.4 Conclusion 49

La résolution du problème du flot optique par variable splitting a été initiée par Zach *et al.* [2007] dans le cadre de la régularisation global $TV - L^1$ (*i.e.* pénalisation de la partie d'adéquation aux données avec la norme L^1 et la régularisation en considérant la variation totale du flot *c.f.* section 1.2.2.2). La même technique est également appliquée au critère $Hubert - L^1$ par Werlberger *et al.* [2009].
Cette méthode consiste à transformer le problème initialement sans contraintes en un problème sous contraintes qui permet de découpler les parties adéquation aux données et régularisation grâce à l'introduction d'une seconde variable. On alterne ensuite la résolution de deux sous-problèmes plus génériques pour lesquels des algorithmes efficaces existent déjà.
Cette approche rencontre un essor remarquable dans la communauté du traitement d'images [Huang *et al.*, 2008; Wang *et al.*, 2008; Afonso *et al.*, 2010; Goldstein et Osher, 2009].

4.1 Reformulation du problème de calcul du flot optique

On se place ici dans le cadre discret. Pour une séquence d'images de dimension $l \times m$, le flot optique $\mathbf{u} = ((u_{i,j})_{\substack{1\leq i\leq l \\ 1\leq j\leq m}}, (v_{i,j})_{\substack{1\leq i\leq l \\ 1\leq j\leq m}})$ est représenté par un vecteur de taille $2lm$. Une approche globale de calcul du flot optique peut s'écrire alors

$$\arg\min_{\mathbf{u}} \mathcal{D}(\mathbf{u}) + \mathcal{R}(\mathbf{u}), \tag{4.1}$$

où $\mathcal{D}(\mathbf{u})$ est un terme d'adéquation aux données quelconque et $\mathcal{R}(\mathbf{u})$ le terme de régularisation choisi.
La méthode de variable splitting consiste à transformer le problème non contraint (4.1) en un problème sous contraintes équivalent

$$\begin{aligned} &\arg\min_{\mathbf{u},\tilde{\mathbf{u}}} \mathcal{D}(\mathbf{u}) + \mathcal{R}(\tilde{\mathbf{u}}), \\ &s.c. \qquad \tilde{\mathbf{u}} = \mathbf{u}, \end{aligned} \tag{4.2}$$

dans lequel on introduit une seconde variable $\tilde{\mathbf{u}}$.

Zach *et al.* [2007] ainsi que Werlberger *et al.* [2009] résolvent ce problème par une méthode de pénalisation quadratique *i.e.* en approchant le problème sous contrainte (4.2) par

$$\mathcal{P}_\beta(\mathbf{u}, \tilde{\mathbf{u}}) = \arg\min_{\mathbf{u},\tilde{\mathbf{u}}} \mathcal{D}(\mathbf{u}) + \mathcal{R}(\tilde{\mathbf{u}}) + \frac{\beta}{2}|\mathbf{u} - \tilde{\mathbf{u}}|^2, \tag{4.3}$$

en faisant croître $\beta > 0$ vers l'infini. Leur stratégie consiste ensuite à alterner la minimisation du problème considéré (4.3) en fixant successivement $\mathbf{u}$ puis $\tilde{\mathbf{u}}$:

$$\mathbf{u}^{k+1} = \arg\min_{\mathbf{u}} \mathcal{D}(\mathbf{u}) + \frac{\beta}{2}|\mathbf{u} - \tilde{\mathbf{u}}^k|^2 \tag{4.4}$$

$$\tilde{\mathbf{u}}^{k+1} = \arg\min_{\tilde{\mathbf{u}}} \mathcal{R}(\tilde{\mathbf{u}}) + \frac{\beta}{2}|\mathbf{u}^{k+1} - \tilde{\mathbf{u}}|^2 \tag{4.5}$$

Remarque 4.1.1 *La démarche consistant à alterner la minimisation de sous-problèmes plus simples est assez classique. On en trouve par exemple dans les méthodes d'optimisation de critères semi-quadratiques [Geman et Reynolds, 1992; Geman et Yang, 1995; Idier, 2001] qui ont été utilisées par Mémin et Pérez [1998] et Aubert* et al. *[1999] pour le calcul du flot optique.*

L'utilisation de β tendant vers l'infini permet d'assurer que le problème (4.3) est équivalent au problème sous contrainte (4.2) et par la suite au problème d'origine (4.1). Cependant, l'usage de grandes valeurs pour β rend le problème $\mathcal{P}_\beta(\mathbf{u}, \tilde{\mathbf{u}})$ difficile à optimiser [Bertsekas, 1996; Bertsekas *et al.*, 1999]. En effet, dans ce cas, la minimisation de la partie pénalisation quadratique prend le pas sur le problème d'origine $\mathcal{D}(\mathbf{u}) + \mathcal{R}(\tilde{\mathbf{u}})$. La convergence vers la solution du problème sous contrainte (4.2) devient alors lente.
Ceci implique notamment que la solution finale de cette méthode dépend de la valeur de β.

Afonso *et al.* [2010] (pour la restauration d'images) ou encore Goldstein et Osher [2009] (pour le débruitage d'image et en "compressive sensing") appliquent l'approche de Bregman [1967] (équivalente à la méthode de Lagrange augmentée [Nocedal et Wright, 1999]) pour développer un algorithme alterné similaire à la méthode (4.4)-(4.5) ne nécessitant pas de condition particulière sur

β afin de garantir la minimisation du problème original (4.1). β sera ici considéré comme un paramètre qui influe sur la vitesse de convergence de la méthode et non pas sur la solution, contrairement à la méthode par pénalisation quadratique.

La méthode de Bregman appliquée au problème général d'approximation du flot optique sera détaillée dans la suite. Le cas particulier de la formulation MCLG sera ensuite traité pour la régularisation par variation totale.

4.2 Méthode de Bregman

La première étape du raisonnement consiste à considérer le problème sous contrainte (4.2) sous la forme équivalente

$$\begin{aligned} &\arg\min_{\mathbf{u},\tilde{\mathbf{u}}} \mathcal{D}(\mathbf{u}) + \mathcal{R}(\tilde{\mathbf{u}}) + \frac{\beta}{2}|\mathbf{u}-\tilde{\mathbf{u}}|^2, \\ &s.c. \qquad \tilde{\mathbf{u}} = \mathbf{u}. \end{aligned} \tag{4.6}$$

Le Lagrangien correspondant à cette formulation du problème du flot optique s'écrit

$$\mathcal{L}(\mathbf{u},\tilde{\mathbf{u}},\lambda) = \mathcal{D}(\mathbf{u}) + \mathcal{R}(\tilde{\mathbf{u}}) + \lambda(\mathbf{u}-\tilde{\mathbf{u}})^T + \frac{\beta}{2}|\mathbf{u}-\tilde{\mathbf{u}}|^2, \tag{4.7}$$

où $\lambda \in \mathbb{R}^{2lm}$ est le multiplicateur de Lagrange et $\beta > 0$ sert cette fois-ci de paramètre de régularisation et n'est pas obligé de tendre vers l'infini.
$\mathcal{L}(\mathbf{u},\tilde{\mathbf{u}},\lambda)$ est le Lagrangien augmenté du problème sous contrainte (4.2).

Remarque 4.2.1 *Le Lagrangien augmenté diffère du Lagrangien classique par la présence du terme de pénalisation quadratique.*
Il correspond en d'autres termes à la méthode de pénalisation quadratique (4.3) *à laquelle on imposerait (par l'ajout du terme de Lagrange) de respecter la condition d'égalité du problème sous contrainte* (4.2)*.*
Le rôle de β est également à noter. En effet, plutôt que d'assurer la convergence de la méthode (comme c'est le cas pour la méthode de pénalisation quadratique), β joue ici sur la rapidité de convergence et permet d'accélérer la méthode de Lagrange classique.

En construisant une suite $(\lambda^k)_{k\in\mathbb{N}}$ convergeant vers λ, la méthode des multiplicateurs consiste à itérativement minimiser le Lagrangien augmenté par rapport à $(\mathbf{u},\tilde{\mathbf{u}})$ en fixant λ^k puis à mettre à jour λ^k [Nocedal et Wright, 1999] :

$$\mathbf{u}^{k+1},\tilde{\mathbf{u}}^{k+1} = \arg\min_{\mathbf{u},\tilde{\mathbf{u}}} \mathcal{L}(\mathbf{u},\tilde{\mathbf{u}},\lambda^k) \tag{4.8}$$

$$\lambda^{k+1} = \lambda^k + \beta(\mathbf{u}^{k+1} - \tilde{\mathbf{u}}^{k+1}) \tag{4.9}$$

La mise à jour du paramètre λ^k selon l'équation (4.9) est justifiée par exemple par [Bertsekas *et al.*, 1999, proposition 4.2.2].

Remarque 4.2.2 *[Bertsekas, 1996, section 2.2.5] Alors que la convergence de la méthode de pénalisation quadratique dépend de la vitesse de croissance de β vers l'infini, la méthode de Lagrange augmentée atteint, dans le pire des cas, une convergence linéaire. Ce taux de convergence peut être supérieur si la valeur initiale λ^0 est proche du multiplicateur de Lagrange λ.*

En remarquant que tout polynôme du second ordre $a|x|^2+bx+c$ peut se ramener sous la forme $a(x+d)^2+k$ où $k=c-\frac{b^2}{4a}$ et $d=-\frac{b}{2a}$ on obtient

$$\lambda(\mathbf{u}-\tilde{\mathbf{u}})^T+\frac{\beta}{2}|\mathbf{u}-\tilde{\mathbf{u}}|^2=\frac{\beta}{2}|\mathbf{u}-\tilde{\mathbf{u}}-d|^2+k, \tag{4.10}$$

$$d=-\frac{\lambda}{\beta}, \tag{4.11}$$

$$k=-\frac{\lambda^2}{2\beta}. \tag{4.12}$$

Le Lagrangien augmenté (4.7) devient alors

$$\mathcal{L}(\mathbf{u},\tilde{\mathbf{u}},\lambda)=\mathcal{D}(\mathbf{u})+\mathcal{R}(\tilde{\mathbf{u}})+\frac{\beta}{2}|\mathbf{u}-\tilde{\mathbf{u}}-d|^2+k. \tag{4.13}$$

Le paramètre k étant constant, il n'intervient pas dans la minimisation de le Lagrangien augmenté (4.13). La méthode de Lagrange augmentée devient alors

$$\mathbf{u}^{k+1},\tilde{\mathbf{u}}^{k+1}=\arg\min_{\mathbf{u},\tilde{\mathbf{u}}}\mathcal{D}(\mathbf{u})+\mathcal{R}(\tilde{\mathbf{u}})+\frac{\beta}{2}|\mathbf{u}-\tilde{\mathbf{u}}-d^k|^2 \tag{4.14}$$

$$d^{k+1}=d^k-(\mathbf{u}^{k+1}-\tilde{\mathbf{u}}^{k+1}). \tag{4.15}$$

Cette approche correspond à l'algorithme de Bregman [1967].

La première équation de l'approche de Bregman (4.14) reste tout de même difficile à résoudre. Comme pour la méthode de pénalisation quadratique, la stratégie consiste à séparer le problème en partie adéquation aux données et partie régularisation. Cette stratégie correspond d'ailleurs à une méthode de Gauss-Seidel par blocs.

La méthode finale proposée consiste alors à mettre à jour alternativement $\mathbf{u}^{k+1}$, $\tilde{\mathbf{u}}^{k+1}$ puis d^{k+1}. On obtient alors une approche similaire à l'algorithme $ADMM$ de [Afonso *et al.*, 2010] :

$$\mathbf{u}^{k+1}=\arg\min_{\mathbf{u}}\mathcal{D}(\mathbf{u})+\frac{\beta}{2}|\mathbf{u}-\tilde{\mathbf{u}}^k-d^k|^2 \tag{4.16}$$

$$\tilde{\mathbf{u}}^{k+1}=\arg\min_{\tilde{\mathbf{u}}}\mathcal{R}(\tilde{\mathbf{u}})+\frac{\beta}{2}|\mathbf{u}^{k+1}-\tilde{\mathbf{u}}-d^k|^2 \tag{4.17}$$

$$d^{k+1}=d^k-(\mathbf{u}^{k+1}-\tilde{\mathbf{u}}^{k+1}). \tag{4.18}$$

Cet algorithme est très proche de la résolution alternée qui découle de la méthode de pénalisation quadratique (4.4)-(4.5). La différence vient de l'ajout du paramètre $d\in\mathbb{R}^{2lm}$ et du fait que β ne doit pas croître vers l'infini pour garantir la résolution du problème non contraint d'origine (4.1).

Remarque 4.2.3 *La parenté des différentes approches "Lagrangiennes" (dont fait partie la méthode de Bregman) est montrée et discutée par plusieurs auteurs tels que [Eckstein et Bertsekas, 1992; Setzer, 2009; Esser, 2009; Tai et Wu, 2009; Chambolle et Pock, 2010].*

La méthode de Bregman tel qu'elle est décrite (4.16)-(4.18) est applicable à n'importe quelle approche globale pour l'approximation du flot optique. La seule contrainte est d'être capable de résoudre les deux premières étapes de manière efficace.
En particulier, la section 4.3 décrit l'application de ce schéma à MCLG. Notons que cette application est valide y compris dans le cas des grands déplacements, contrairement à ce qu'ont affirmé Steinbruecker *et al.* [2009].

4.3 Application à la méthode MCLG

Dans la suite, en revenant à la combinaison des régularisations locale et globale, les fonctions $\mathcal{D}$ et $\mathcal{R}$ utilisées sont explicitées. Les méthodes de résolution des sous-problèmes (4.16) et (4.17) sont alors décrites.

4.3.1 Résolution de la partie adéquation aux données

Cette section concerne la résolution de la partie attachée aux données (4.16) de la méthode de Bregman. La fonction $\mathcal{D}$ traduisant l'hypothèse de régularité locale du flot étudiée ici est :

$$\mathcal{D}(\mathbf{u}) = \alpha \sum_{i=1}^{l} \sum_{j=1}^{m} \mathcal{G} * \left(I(\cdot + \mathbf{u}_{i,j}) - I(\cdot)\right)^2 \tag{4.19}$$

où $\mathcal{G}$ est un noyau discret de convolution et "$\cdot$" est la variable muette de l'opération de convolution. On notera que le paramètre de réglage $\alpha > 0$ est cette fois-ci porté sur la partie adéquation aux données. En effet, cela permet, dans le cas de la régularisation de la variation totale du champ de vecteurs, de retrouver le modèle de débruitage de Rudin, Osher, et Fatemi [1992] pour la partie régularisation (4.17).

En reprenant la version modifiée de CLG (2.9) ainsi que les notations du chapitre 2, le terme d'adéquation aux données avec warping est noté

$$\mathcal{D}^n(\mathbf{u}) = \alpha \sum_{i=1}^{l} \sum_{j=1}^{m} \mathcal{G} * \left(\mathbf{u}_{i,j} \bar{\nabla}_3 I^{n^T} \bar{\nabla}_3 I^n \mathbf{u}_{i,j}{}^T\right). \tag{4.20}$$

Cette manipulation permet de combiner un schéma de warping justifié avec la méthode de Bregman. En effet, l'algorithme proposé s'écrit

$$\mathbf{u}^{n+1,k+1} = \arg\min_{\mathbf{u}} \mathcal{D}^n(\mathbf{u}) + \frac{\beta}{2}|\mathbf{u} - \tilde{\mathbf{u}}^{n+1,k} - d^{n+1,k}|^2 \tag{4.21}$$

$$\tilde{\mathbf{u}}^{n+1,k+1} = \arg\min_{\tilde{\mathbf{u}}} \mathcal{R}(\tilde{\mathbf{u}}) + \frac{\beta}{2}|\mathbf{u}^{n+1,k+1} - \tilde{\mathbf{u}} - d^{n+1,k}|^2 \tag{4.22}$$

$$d^{n+1,k+1} = d^{n+1,k} - (\mathbf{u}^{n+1,k+1} - \tilde{\mathbf{u}}^{n+1,k+1}). \tag{4.23}$$

où n désigne la récursion du warping et k celle de l'algorithme de Bregman.

La résolution de la partie adéquation aux données (4.21) de cette méthode revient alors à l'inversion d'un système 2×2 en chaque pixel.
En effet, $\mathbf{u}^{n+1,k+1}$ s'obtient en résolvant

$$\arg\min_{\mathbf{u}} \sum_{i=1}^{l} \sum_{j=1}^{m} \alpha \mathcal{G} * \left(\mathbf{u}_{i,j} \bar{\nabla}_3 I^{nT} \bar{\nabla}_3 I^n \mathbf{u}_{i,j}{}^T \right) + \frac{\beta}{2} (\mathbf{u}_{i,j} - \tilde{\mathbf{u}}_{i,j}^{n+1,k} - d_{i,j}^{n+1,k})^2. \quad (4.24)$$

Ce problème est séparable. Ainsi, la minimisation de ce critère s'effectue en annulant sa dérivée en tout pixel. Il en découle

$$\begin{pmatrix} u_{i,j}^{n+1,k+1} \\ v_{i,j}^{n+1,k+1} \end{pmatrix} = \begin{pmatrix} \mathcal{J}_{i,j}^{xx} + \beta & \mathcal{J}_{i,j}^{xy} \\ \mathcal{J}_{i,j}^{xy} & \mathcal{J}_{i,j}^{yy} + \beta \end{pmatrix}^{-1} \begin{pmatrix} -\mathcal{J}_{i,j}^{xt} + \beta(\tilde{u}_{i,j}^{n+1,k} + d_{i,j}^{n+1,k}) \\ -\mathcal{J}_{i,j}^{yt} + \beta(\tilde{v}_{i,j}^{n+1,k} + d_{i,j}^{n+1,k}) \end{pmatrix} \quad (4.25)$$

où

$$\mathcal{J}^{\diamond\star} = \alpha \mathcal{G} * (\partial_\diamond I^n \partial_\star I^n) \qquad \diamond, \star \in \{x, y, t\}. \quad (4.26)$$

La première étape des itérations de la méthode de Bregman est ainsi résolue.

4.3.2 Résolution de la partie régularisation

Comme au chapitre 3, nous choisissons ici la pénalisation de la variation totale du flot optique. On notera pour cela

$$\mathcal{R}(\mathbf{u}) = \sum_{i=1}^{l} \sum_{j=1}^{m} |\nabla u_{i,j}| + |\nabla v_{i,j}| \quad (4.27)$$

La minimisation du critère de régularisation (4.22) relatif à $\tilde{\mathbf{u}}$ correspond alors au débruitage de $\mathbf{u}^{k+1} - d^k$ par régularisation de la variation totale tel que décrit par Rudin, Osher, et Fatemi [1992].
Plusieurs méthodes de résolution efficaces de ce modèle on été développées. Notamment, Chambolle [2004] propose un algorithme globalement convergent. Zach *et al.* [2007] ou encore Afonso *et al.* [2010], utilisent cette méthode pour la résolution de la partie régularisation (4.22).
Werlberger *et al.* [2009] quant à eux s'inspirent de cette méthode pour produire un procédé itératif pour la minimisation de la variation totale approchée par la fonction de Huber [1973]

$$|x| \approx |x|_\varepsilon = \begin{cases} |x| - \frac{\varepsilon}{2} & \text{si } |x| > \varepsilon, \\ \dfrac{|x|^2}{2\varepsilon} & \text{sinon.} \end{cases} \quad (4.28)$$

C'est cette dernière approche qui est choisie ici car elle permet de limiter l'effet "d'escalier" (*staircasing effect*) caractéristique des solutions obtenues par régulari-

sation de la variation totale du flot [Werlberger *et al.*, 2009]. On obtient ainsi

$$\tilde{w}^{k+1} = w^{k+1} - d^k + \frac{1}{\beta}\mathrm{div}(p_w^k) \qquad w \in \{u, v\}, \tag{4.29}$$

$$p_w^{k+1} = \frac{p_w^k + \tau(\nabla\tilde{w}^{k+1} - \varepsilon p_w^k)}{\max\{1, |p_w^k + \tau(\nabla\tilde{w}^{k+1} - \varepsilon p_w^k)|\}} \qquad \tau = \frac{1}{4+\varepsilon}. \tag{4.30}$$

4.3.3 Parallélisation

Comme déjà stipulé précédemment, le traitement de la partie adéquation aux données implique la résolution d'un système linéaire 2×2 en tout point indépendamment des autres.

De même, les opérations de minimisation de la partie régularisation de la méthode de Bregman sont indépendantes en chaque pixel.
La parallélisation à grain fin dans un schéma SIMD de l'algorithme de Bregman appliqué à la minimisation du critère MCLG est donc naturelle et aucune manipulation de type coloration de mailles n'est nécessaire dans les méthodes de résolutions choisies ici.
L'inclusion dans une approche de parallélisation à mémoire partagée est également possible comme décrit à l'annexe C.

Remarque 4.3.1 *D'autres méthodes de résolution pour l'étape* (4.22) *qui traite la partie régularisation de l'équation peuvent être envisagées. Le choix est guidé ici par le compromis offert par la méthode de Welberger* et al. *entre qualité du résultat, indépendance des opérations et facilité de mise en œuvre.*
L'approche de Goldstein et Osher [2009] semble être une bonne alternative car elle permettrait la régularisation à l'aide de la variation totale aussi bien dans sa version isotrope que anisotrope. Cependant, sa parallélisation entraînerait l'emploi des même techniques de réordonancement des opérations que celles décrites à la section précédente. Cette méthode n'a pas été testée au cours de cette thèse.

4.4 Conclusion

Ce chapitre a permis de proposer un autre mode de résolution du problème par l'utilisation du paradigme de variable splitting. Ces travaux sont à notre connaissance les premiers à appliquer la méthode split Bregman au problème d'estimation du flot optique.

Après avoir présenté les fondements de cette technique, nous l'avons appliquée à la méthode MCLG. Cette technique permet par construction de se ramener à la résolution d'une succession de sous problèmes pour lesquels il existe des algorithmes simples de résolution. Ceux-ci ont l'avantage de ne comporter que des opérations décorrélées donc facilement parallélisable.

La prochaine partie donnera des précisions sur la mise en œuvre des différents algorithmes développés au chapitre 3 et 4. Une comparaison de leurs performances sera ensuite faite, aussi bien du point de vue qualitatif que numérique.

Chapitre 5

Évaluation du parallélisme des méthodes développées

Sommaire

5.1 Détails de mise en œuvre 51
5.1.1 Schéma multi-niveaux 52
5.1.2 Construction des systèmes 53
5.1.3 Gestion des bords 53
5.1.4 Structure algorithmique 54
5.1.5 Décompte des opérations 54
5.2 Mise en œuvre CPU/OpenMP 56
5.2.1 Modèle de parallélisation utilisé 56
5.2.2 Sélection des versions comparées 56
5.2.3 Résultats 57
5.3 Mise en œuvre GPU/CUDA 63
5.3.1 Le modèle de programmation sur GPU 63
5.3.2 Mesure de performances 65
5.4 Conclusion 67

Cette partie a pour vocation d'évaluer les différents algorithmes développés dans les chapitres précédents. La première partie donne quelques détails de mise en œuvre communs aux différents algorithmes proposés. Il s'ensuit une étude détaillée de leurs complexités numériques respectives.
Pour finir, on procède à l'évaluation du parallélisme des algorithmes développés aussi bien sur CPU multi-cœurs que sur processeur graphique.

5.1 Détails de mise en œuvre

Le chapitre 2 a permis de distinguer deux familles d'algorithmes pour la gestion de la partie adéquation aux données :

CLG **:** la méthode CLG de Bruhn *et al.*,

$MCLG$ **:** notre méthode MCLG.

Les chapitres 3 et 4 ont défini deux méthodes de résolution différentes :

CV **:** calcul variationnel,

VS **:** variable splitting.

Dans le cadre des méthodes de résolution par calcul variationnel, quatre manières de gérer la régularisation de la variation totale ont été développées :

ISO_{4ppv} **:** gestion classique de l'approximation isotrope (faisant intervenir les quatre plus proches voisins du point considéré),

$ANISO_{4ppv}$ **:** gestion classique de l'approximation anisotrope (faisant intervenir les quatre plus proches voisins du point considéré),

ISO_{8ppv} **:** approche proposée pour l'approximation isotrope (faisant intervenir les huit plus proches voisins du point considéré),

$ANISO_{8ppv}$ **:** approche proposée pour l'approximation anisotrope (faisant intervenir les huit plus proches voisins du point considéré).

Ce chapitre va ainsi s'intéresser à l'évaluation de neuf algorithmes :
- $CLG^{CV}_{ISO_{4ppv}}$, $CLG^{CV}_{ANISO_{4ppv}}$, $CLG^{CV}_{ISO_{8ppv}}$, $CLG^{CV}_{ANISO_{8ppv}}$,
- $MCLG^{CV}_{ISO_{4ppv}}$, $MCLG^{CV}_{ANISO_{4ppv}}$, $MCLG^{CV}_{ISO_{8ppv}}$, $MCLG^{CV}_{ANISO_{8ppv}}$,
- $MCLG^{VS}$.

Les neuf algorithmes précités sont développés sur CPU à l'aide d'un code écrit en C et parallélisé à l'aide d'openMP [Chapman *et al.*, 2008], et la version $MCLG^{CV}_{iso_{4ppv}}$ a été portée sur GPU à l'aide d'un code C/CUDA [Nvidia, 2007]. On utilise une interface Python [Pilgrim, 2004] pour la lecture des données et l'affichage des résultats.

Cette partie commence par décrire les aspects communs de ces différents algorithmes. Nous présenterons ensuite les points spécifiques de telle ou telle version.

5.1.1 Schéma multi-niveaux

Les différents solveurs développés sont inscrits dans un schéma multi-échelles. Cette technique connue en traitement d'images sous le nom de "*coarse to fine*" correspond en fait à une approche multigrille unidirectionnelle (ou encore "cascadic multigrid" [Bornemann et Deuflhard, 1996]).

Pour ce faire, une pyramide d'images est construite à partir de la séquence étudiée (*c.f.*FIGURE 5.1). En partant de la grille la plus grossière, la stratégie consiste ensuite à résoudre le problème à chaque niveau et à utiliser la solution obtenue pour initialiser la résolution au niveau plus fin suivant (*c.f.*FIGURE 5.2).

Le facteur de réduction inter-niveaux de grille est arbitraire. Il est fixé généralement à 0.8 au court des expériences menées dans ce chapitre. Nos expériences montrent que cette valeur permet d'obtenir les meilleurs résultats. D'autre part, les transferts inter-grilles sont fait par interpolation bilinéaire. On impose également qu'aucune des dimensions de la grille la plus grossière ne soit inférieure à 16.

Cette technique permet de fournir une solution initiale convenable au solveur itératif. Cette initialisation est d'autant meilleure que les grands mouvements de la

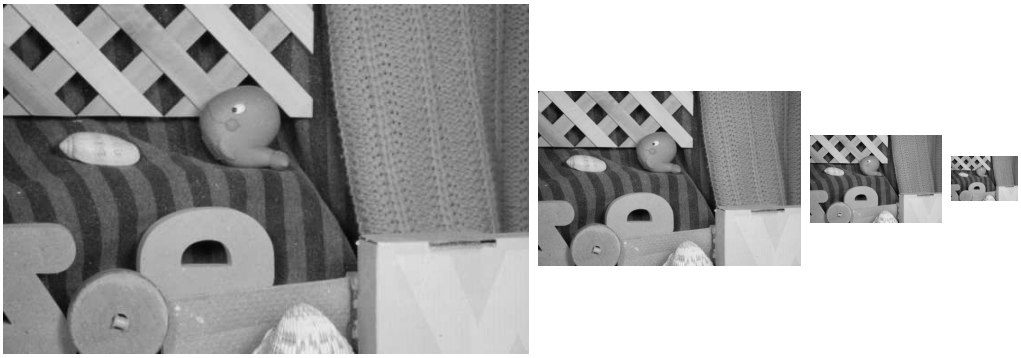

FIGURE 5.1 – Exemple de pyramide dyadique.

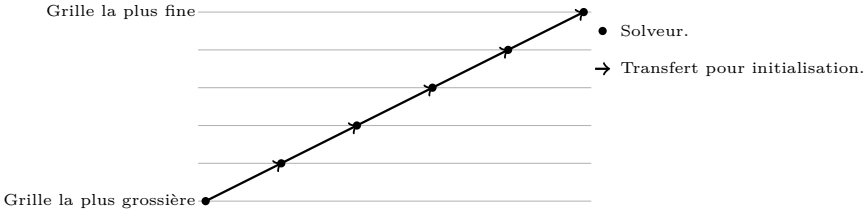

FIGURE 5.2 – Schéma de résolution multi-niveau (ou multigrille unidirectionnel).

grille fine projetée sur les grilles grossières sont réduit à ces échelles et donc plus faciles à estimer.
De plus, comme le montre la FIGURE 5.3, la proportion des calculs effectués sur les grilles grossières est raisonnable comparée à celle des calculs effectués sur la grille la plus fine.

5.1.2 Construction des systèmes

Les gradients spatiaux des images sont approchés par un schéma de différences finies à l'ordre trois à l'aide du noyau $[\frac{1}{12}, \frac{-8}{12}, 0, \frac{8}{12}, \frac{-1}{12}]$.

En ce qui concerne les gradients des composantes du flot optique, ils sont approchés par un schéma centré dans le cadre du calcul variationnel et à l'ordre un (avant ou arrière) pour la résolution par variable splitting.

D'autre part, le noyau de convolution étant choisi gaussien, il est séparable. Ceci permet d'économiser des opérations en décomposant la convolution bidimensionnelle en deux convolutions unidirectionnelles horizontale puis verticale.

5.1.3 Gestion des bords

Pour éviter d'imposer des condition aux bords artificielles, les opérations de convolution seront effectuées uniquement à l'intérieur du domaine avec une marge égale au rayon du noyau de convolution.

Ce problème se pose aussi lors du warping des images : lorsque les pixels sortent du domaine image, l'intensité des points concernés est extrapolée à l'endroit où le

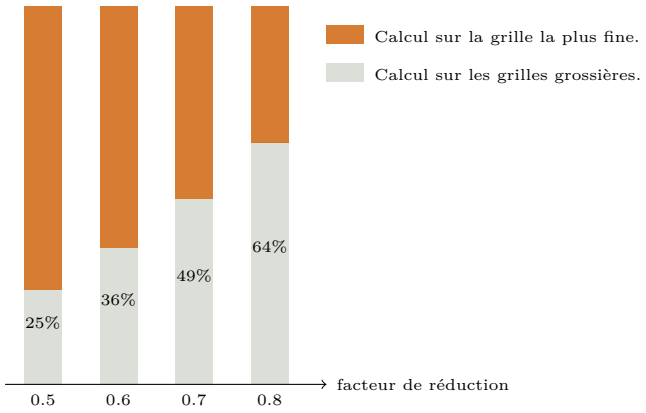

FIGURE 5.3 – Proportion des calculs effectués sur les grilles grossières dans un schéma multi-échelles unidirectionnel. Le nombre de niveaux de la pyramide correspond au maximum possible sous condition que la grille la plus grossière soit de dimension au moins 16×16.

pixel sort. Pour ne pas introduire d'erreur lors de cette opération, l'idée consiste à n'imposer en ces points que la régularité globale du champ de vecteurs. Pour ce faire, les coefficients de la partie adéquation aux données sont annulés.

5.1.4 Structure algorithmique

En définissant N_{niv} le nombre de niveaux de la pyramide, N_{warp} le nombre de re-linéarisations du système et finalement N_{iter} le nombre de récursions du solveur, la structure générale des neuf algorithmes développés est

Algorithme 1

construction d'une pyramide d'images à N_{niv} niveaux,
pour niv **allant de** 1 **à** N_{niv} **:**
  prolonger la solution de la grille précédente (sauf pour la plus grossière),
  pour warp **allant de** 1 **à** N_{warp} **:**
    interpoler l'image à l'aide de la solution courante,
    construire les coefficients du système,
    pour iter **allant de** 1 **à** N_{iter} **:**
      appliquer une itération de résolution (dépend de la méthode choisie).

5.1.5 Décompte des opérations

L'évaluation des différents algorithmes développés passe par le décompte des opérations à virgule flottante effectuées.
Ainsi, en chaque pixel et à chaque niveau de pyramide, le nombre $flop_{solv}$ d'opéra-

Tableau 5.1 – Détail des quantités déterminant le nombre d'opérations par pixel pour chaque algorithme mis en œuvre.

	$flop_{iter}$	$flop_{syst}$	$flop_{prol}$
$CLG^{CV}_{ISO_{4ppv}}$	81	$69 + 16 \times R$	32
$CLG^{CV}_{ANISO_{4ppv}}$	90		
$CLG^{CV}_{ISO_{8ppv}}$	108		
$CLG^{CV}_{ANISO_{8ppv}}$	134		
$MCLG^{CV}_{ISO_{4ppv}}$	59	$73 + 16 \times R$	
$MCLG^{CV}_{ANISO_{4ppv}}$	68		
$MCLG^{CV}_{ISO_{8ppv}}$	82		
$MCLG^{CV}_{ANISO_{8ppv}}$	108		
$MCLG^{VS}$	120	$75 + 16 \times R$	180

tions nécessaires à la résolution est

$$flop_{solv} = N_{warp} \times (flop_{syst}(R) + N_{iter} \times flop_{iter}), \tag{5.1}$$

où

- $flop_{syst}$ est le nombre d'opérations dédiées à la construction du système, cette quantité dépend du rayon R du noyau de convolution,
- $flop_{iter}$ est le nombre d'opérations effectuées au cours d'une itération de solveur.

Remarque 5.1.1 *Le rayon du noyau de convolution R dépend de l'écart-type de la gaussienne σ utilisée. On choisit $R = 3\sigma$.*

Pour chaque point d'une grille (sauf la plus grossière), le coup du calcul, $flop_{prol}$, des valeurs initiales des différentes inconnues du système est également pris en compte. Les nombres $flop_{iter}$, $flop_{syst}$ et $flop_{prol}$ sont propres à l'algorithme choisi. Le tableaux 5.1.5 détaille ces quantités.
Il faut également compter le coût de construction de la pyramide d'images $flop_{pyr}$. Il correspond à 32 opérations par point de grille à chaque niveau (sauf au niveau le plus fin). Ainsi, en définissant N^i_{pix}, le nombre de pixels à traiter au niveau de pyramide i (avec la convention que le niveau 1 est le plus grossier et N_{niv} le plus fin), est :

$$flop_{pyr} = 32 \times \sum_{i=1}^{N_{niv}-1} N^i_{pix}. \tag{5.2}$$

Finalement, le nombre total d'opérations $flop_{total}$ est obtenu à l'aide de la formule

$$flop_{total} = flop_{pyr} + N^1_{pix} \times flop_{solv} + \sum_{i=2}^{N_{niv}} N^i_{pix} \times (flop_{solv} + flop_{prol}). \tag{5.3}$$

5.2 Mise en œuvre CPU/OpenMP

Un processeur bi-Xeon comportant douze unités de calculs cadencées à 2.93Ghz a été utilisé pour cette partie des expérimentations. Des mesures sur un produit matrice×matrice de la bibliothèque BLAS (sgemm) [Dongarra *et al.*, 2002] indiquent que son pic de performances est de 141.5Gflops en simple précision.

Remarque 5.2.1 *La qualité des résultats ne souffre pas de l'utilisation de la simple précision à la place de la double précision (l'erreur absolue moyenne mesurée, entre la solution en simple précision et celle en double précision, sur l'ensemble des séquences utilisées pour l'évaluation au chapitre 6 est de 0.04 pixel). Ainsi, pour assurer un maximum de rapidité d'exécution, les calculs sont effectués en simple précision. De plus, ceci garantira une comparaison objective des performances des GPU qui ne permettent, pour leurs majorité, que des calculs en simple précision.*

5.2.1 Modèle de parallélisation utilisé

L'algorithme générique 5.1.4 repose sur une succession de routines composées d'opérations décorrélées. En effet, la construction de la pyramide d'images, les transferts inter-grille des solutions et le warping des images est basé sur l'interpolation bilinéaire et peut être effectuée pour chaque pixel de manière parallèle. De même, pour la construction des coefficients du système repose sur les opérations de différentiation et de convolution qui sont elles aussi décorellées en chaque pixel. D'autre part, les itérations des solveurs ont été construites de manière à permettre également un traitement parallèle : pour la résolution par calcul variationnel les opérations ont été réordonnées par une technique de coloriage de mailles (chapitre 3) et pour la résolution par variable splitting, la parallélisation est naturelle.

Pour exploiter le parallélisme des routines précitées, on parcourra les points du domaine image à l'aide d'une boucle parallèle. La bibliothèque openMP [Chapman *et al.*, 2008] permet ceci en plaçant des directives avant les boucles à paralléliser de la forme :

```
#pragma parallel for private(...) shared(...)
```

où l'on doit prendre soin de déclarer les variables privées et celles qui seront partagées par les différents threads.

5.2.2 Sélection des versions comparées

Pour simplifier l'analyse on se limite à la comparaison de trois algorithmes :
- la résolution par calcul variationnel de CLG et MCLG avec régularisation par variation totale isotrope classique ($CLG^{CV}_{ISO_{4ppv}}$ et $MCLG^{CV}_{ISO_{4ppv}}$),
- la résolution de MCLG par variable splitting ($MCLG^{VS}$).

Les paramètres utilisés lors des différentes expériences de cette section sont les paramètres par défaut (*c. à d.* ceux qui permettent le meilleur compromis rapidité/précision), à savoir :

- N_{niv} = maximum possible pour un facteur de réduction de 0.8 et une grille grossière de taille au moins 16×16,
- N_{warp} =5,
- $R = 2$,
- $N_{iter} = 100$ pour la résolution par calcul variationnel et 10 par variable splitting.

Au cours des expériences suivantes, plusieurs quantités seront mesurées :

- le temps de calcul T,
- le nombre d'opérations par seconde,

$$Flops = \frac{flop_{total}}{T} \qquad \text{où } flop_{total} \text{ est mesuré à l'aide de (5.3),} \tag{5.4}$$

- l'accélération offerte par un nombre p de processeurs :

$$A_p = \frac{T_1}{T_p} \qquad \text{où } T_i \text{ est le temps mesuré pour } i \text{ processeurs,} \tag{5.5}$$

  qui permet de comparer le code parallèle à sa version séquentielle ;
- l'efficacité de l'algorithme parallèle :

$$E_p = \frac{A_p}{p}. \tag{5.6}$$

  qui donne une indication sur le taux d'utilisation des processeurs disponibles.

5.2.3 Résultats

Nous comparons dans cette partie, à travers différentes expériences, MCLG à CLG ainsi que la résolution par calcul variationel et celle par variable splitting. On considère pour cela les algorithmes $MCLG^{CV}_{ISO_{4ppv}}$, $CLG^{CV}_{ISO_{4ppv}}$ et $MCLG^{VS}$.

5.2.3.1 Nombre de threads fixe et taille d'images variable

La première expérience illustrée à la FIGURE 5.4 consiste à mesurer les performances offertes par douze threads pour des dimensions d'images allant de 300×300 à 1900×1900.

Comparons tout d'abord MCLG et CLG à travers l'étude des courbes de $MCLG^{CV}_{ISO_{4ppv}}$, $CLG^{CV}_{ISO_{4ppv}}$. L'apport de notre approche est net. On constate donc qu'en plus d'être une correction théorique de la méthode de Bruhn *et al.*, le développement MCLG offre un gain en temps de calcul de l'ordre de 30% par rapport à CLG.

Outre une économie en nombre d'opérations de calcul (voir le tableau 5.1.5), la méthode $MCLG^{CV}_{ISO_{4ppv}}$ semble mieux bénéficier de la gestion automatique de la

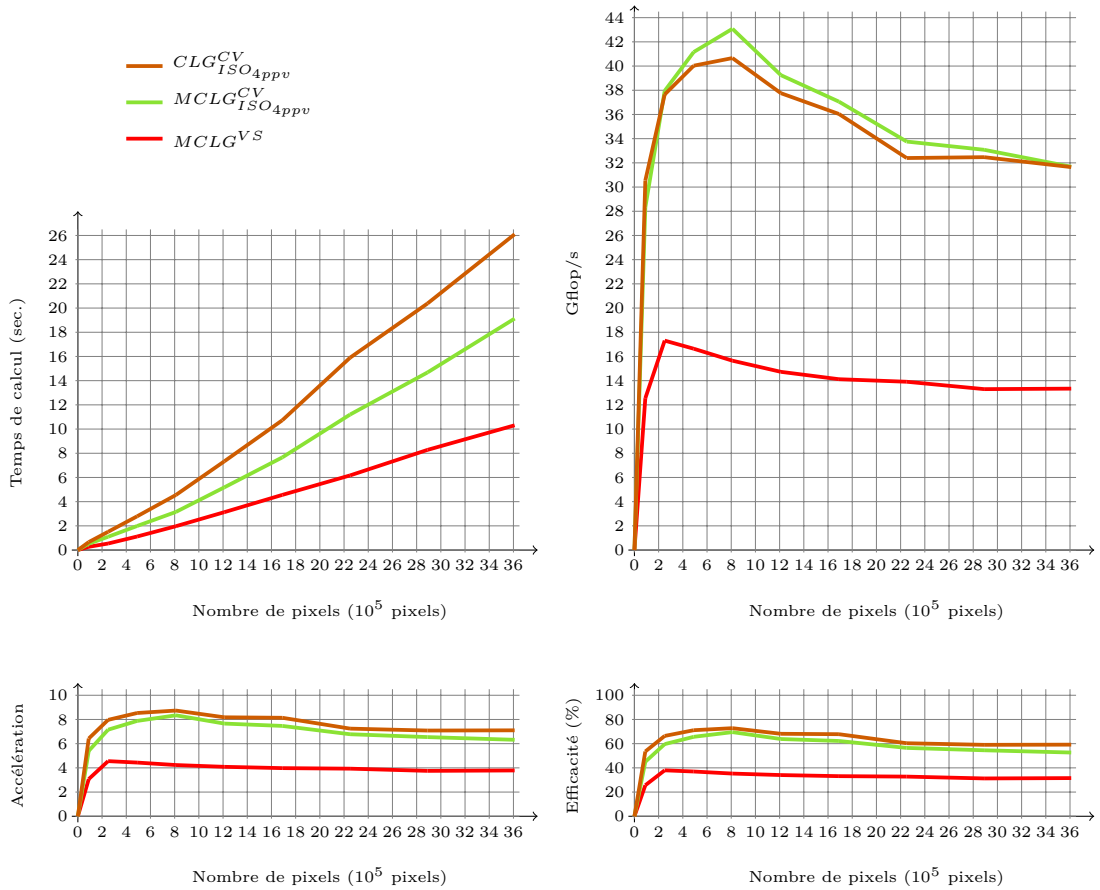

FIGURE 5.4 – Mesure des performances des algorithmes avec 12 threads et taille d'images variable (de 300×300 à 1900×1900). **Haut gauche :** temps de calcul. **Haut droit :** nombre d'opérations par seconde. **Bas gauche :** Accélération. **Bas droit :** Efficacité.

mémoire cache que $CLG^{CV}_{ISO_{4ppv}}$. Ceci se voit en examinant le nombre d'opérations effectuées par seconde : ce nombre est plus important pour MCLG, notamment pour les images de dimensions 700×700.

En revanche, $CLG^{CV}_{ISO_{4ppv}}$ conduit à une parallélisation plus efficace que $MCLG^{CV}_{ISO_{4ppv}}$. En effet, on observe pour cette méthode une accélération proche de 9 par rapport au code séquentiel pour certaines dimensions d'images ce qui lui permet d'atteindre 70% d'efficacité.

En ce qui concerne la comparaison de la résolution par variable splitting à celle par calcul variationnel, on voit que la première est presque 2 fois plus rapide que la seconde. Cela malgré le fait que ce soit la méthode qui nécessite le plus grand nombre d'opérations par itération de solveur $flop_{iter}$ et construction de système $flop_{syst}$. Ceci s'explique par le fait que cette méthode nécessite bien moins d'itérations N_{iter} que la résolution par calcul variationnel (10 fois moins) pour obtenir une solution

correcte.

D'autre part, on observe que la résolution par variable splitting est celle qui effectue le moins d'opérations par seconde. Sans doute parce que l'algorithme entraîne peu de connections entre les données. Ainsi, les données mises en mémoire cache ne sont pas réutilisées. Il en résulte une faible accélération par rapport au code séquentiel et une efficacité moindre.
En revanche, grâce au réordonnancement des opérations de la résolution par calcul variationnel, plusieurs points d'une même couleur utiliseront une donnée commune mise en cache (voire FIGURES 3.1 et 3.2), d'où une meilleure exploitation de la mémoire cache, ce qui offre une accélération importante et une meilleur efficacité.

Les courbes représentant le nombre d'opérations par seconde présentent toutes trois un point à partir duquel elles se mettent à décroître après une forte croissance. Ceci illustre le goulot d'étranglement de Von Neumann [Backus, 1978] qui est dû à un débit trop faible du transfert des données entre la mémoire et le processeur, ce qui fait que ce dernier se met à l'arrêt en attendant l'arrivée des données qu'il doit traiter.
Analysons plus en détail ce phénomène pour la méthode $MCLG^{CV}_{ISO_{4ppv}}$. Cet algorithme nécessite le stockage de sept tableaux de données : les deux composantes du champ de vecteurs (u et v) ainsi que les coefficients provenant de la partie adéquation aux données ($\mathcal{J}^{xx}, \mathcal{J}^{xy}, \mathcal{J}^{yy}, \mathcal{J}^{xt}$ et $\mathcal{J}^{yt}$ en employant les notations (4.26)).
Ainsi, pour une image comportant N pixels, la quantité de mémoire utilisées par cette méthode est de $4 \times N \times 7$ octets (car on utilise 7 tableaux d'éléments à virgule flottante en simple précision, codé sur 4 octets) sur le niveau de grille le plus fin.
D'après cette courbe, $MCLG^{CV}_{ISO_{4ppv}}$ atteint son pic de performance pour $8 \cdot 10^5$ pixels. Donc, la mémoire nécessaire au calcul sur la grille la plus fine est de 22.4Mo, ce chiffre correspond à la taille du cache L3 du processeur qui est de 2×12Mo. Ces chiffres confirment notre interprétation de ce phénomène.

Une manière de contourner ce genre de situation est de mettre en œuvre une résolution par bloc, en essayant d'exploiter au maximum le cache disponible.

5.2.3.2 Taille d'images fixe et nombre de threads variable

On cherche par cette expérience à observer le comportement des algorithmes lors de l'augmentation du nombre de processeurs dédiés au calcul. On veut ainsi évaluer à quel point ces algorithmes se prêtent à la parallélisation, ou, en d'autres termes leur extensibilité ("*scalability*").
Pour ce faire, la taille des images traitées est fixée à 700×700 pixels, puis des mesures de temps sont effectuées en utilisant un nombre d'unités de calcul variant de 1 à 12.

Pour tous les nombres de threads impliqués, la résolution par variable splitting

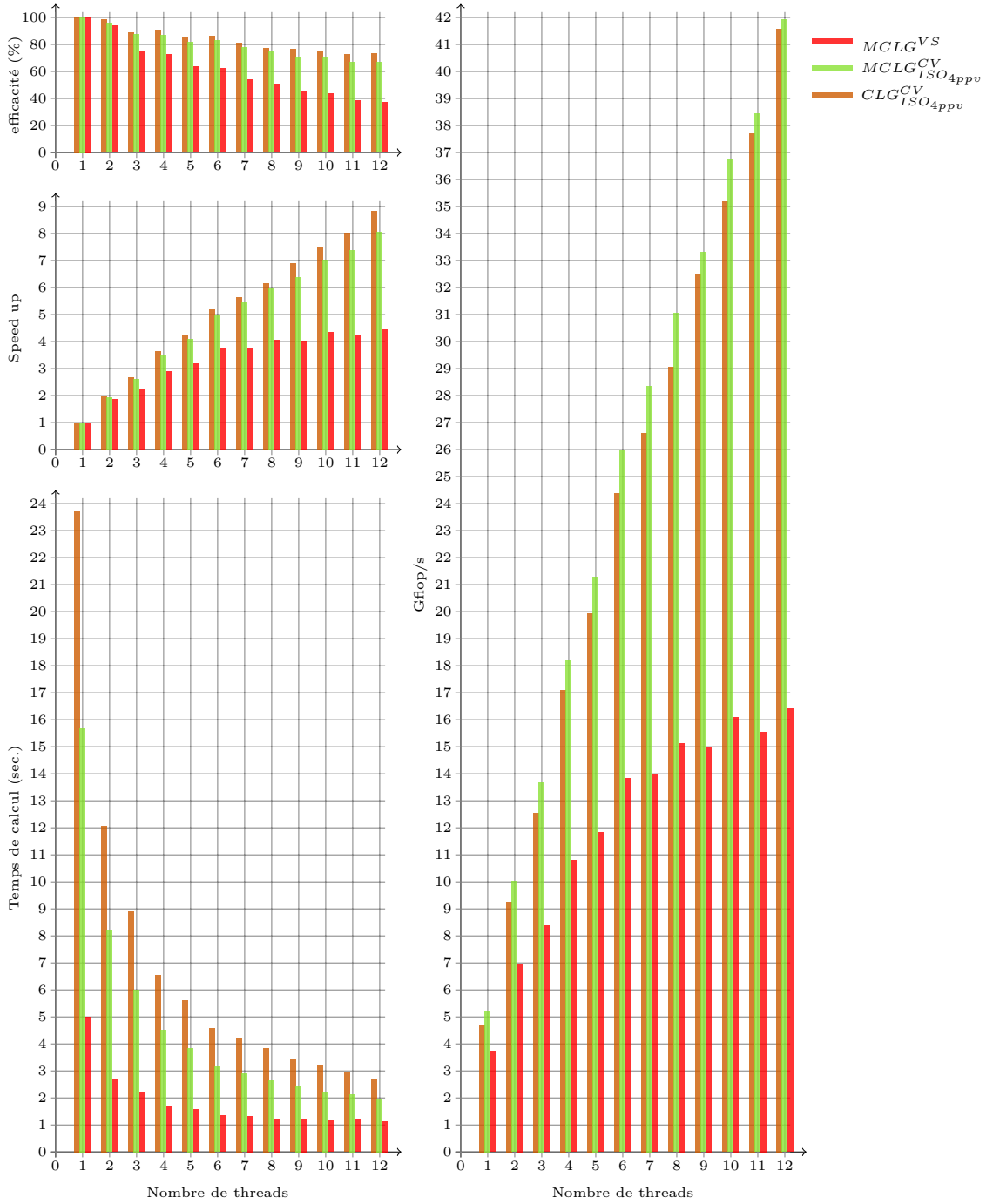

FIGURE 5.5 – Mesure des performances des différentes méthodes développées en faisant varier le nombre de threads avec une dimension d'images fixe de 700×700. **Haut gauche :** efficacité. **Milieu gauche :** accélération. **Bas gauche :** temps de calcul. **droite :** nombre d'opérations flottantes par seconde.

offre les temps de calcul les plus courts : jusqu'à sept threads, ce temps est respectivement quatre et trois fois plus court que pour CLG et MCLG résolu par calcul variationnel.
Cependant, le nombre d'opérations par seconde le plus grand est obtenu par la résolution par calcul variationnel avec un léger avantage pour MCLG et cela pour les raisons exposées à section 5.2.3.1.

Il est remarquable que, alors que le nombre d'opérations par seconde des méthodes de résolution par calcul variationnel croît en fonction du nombre de threads, celui de l'approche par variable splitting évolue très peu à partir de 6 threads. Ce phénomène se retrouve également pour l'accélération. Il en résulte une décroissance franche de l'efficacité de $MCLG^{VS}$.

On peut ainsi considérer que la méthode de résolution par variable splitting atteint ses meilleurs performances lorsque 6 threads sont utilisés et qu'il n'est plus nécessaire à partir de ce nombre de lui allouer plus de ressources de calcul.

5.2.3.3 Volume de données constant par thread

L'extensibilité d'un algorithme est fonction de la quantité de données que chaque thread doit traiter au cours du calcul. Ainsi, s'il y a trop peu de données comparé au nombre de threads, certains d'entre eux seront moins actifs durant les calculs, ce qui nuit à l'efficacité du solveur.
Pour éviter ce genre de situation, et confirmer les résultats de l'expérience précédente, l'idée du test suivant est de fournir une quantité de données constante par processeur. Cela a consisté dans notre cas à mesurer les performances des algorithmes sur des problèmes de tailles $300 \times 300 \times p$ où p est le nombre de threads utilisés, de sorte que chaque thread ait en moyenne 300×300 points à traiter.

On retrouve dans la FIGURE 5.6 les même comportements que ceux observés à taille d'images constante pour le nombre d'opérations par seconde et l'accélération (*c.f.*FIGURE 5.5) : c'est à dire, un pic de performance pour $MCLG^{VS}$ obtenu avec 6 threads.

En ce qui concerne le temps de calcul et l'efficacité, un résultat idéal serait constant en fonction du nombre de thread. Un tel résultat est impossible à obtenir car l'augmentation du volume des données à traiter entraîne toujours davantage de transfert. Ainsi, moins ces courbes sont croissantes, meilleur est le résultat. On remarquera donc le très bon comportements des algorithmes de résolution par calcul variationnel. En effet, leurs efficacités respectives décroît nettement moins que pour la méthode de variable splitting.

5.2.3.4 Synthèse

Bien que nécessitant un nombre d'opérations par itération plus important, les expériences menées précédemment montrent que la résolution par variable splitting donne les meilleurs temps de calculs. Cela est dû au faible nombre d'itérations

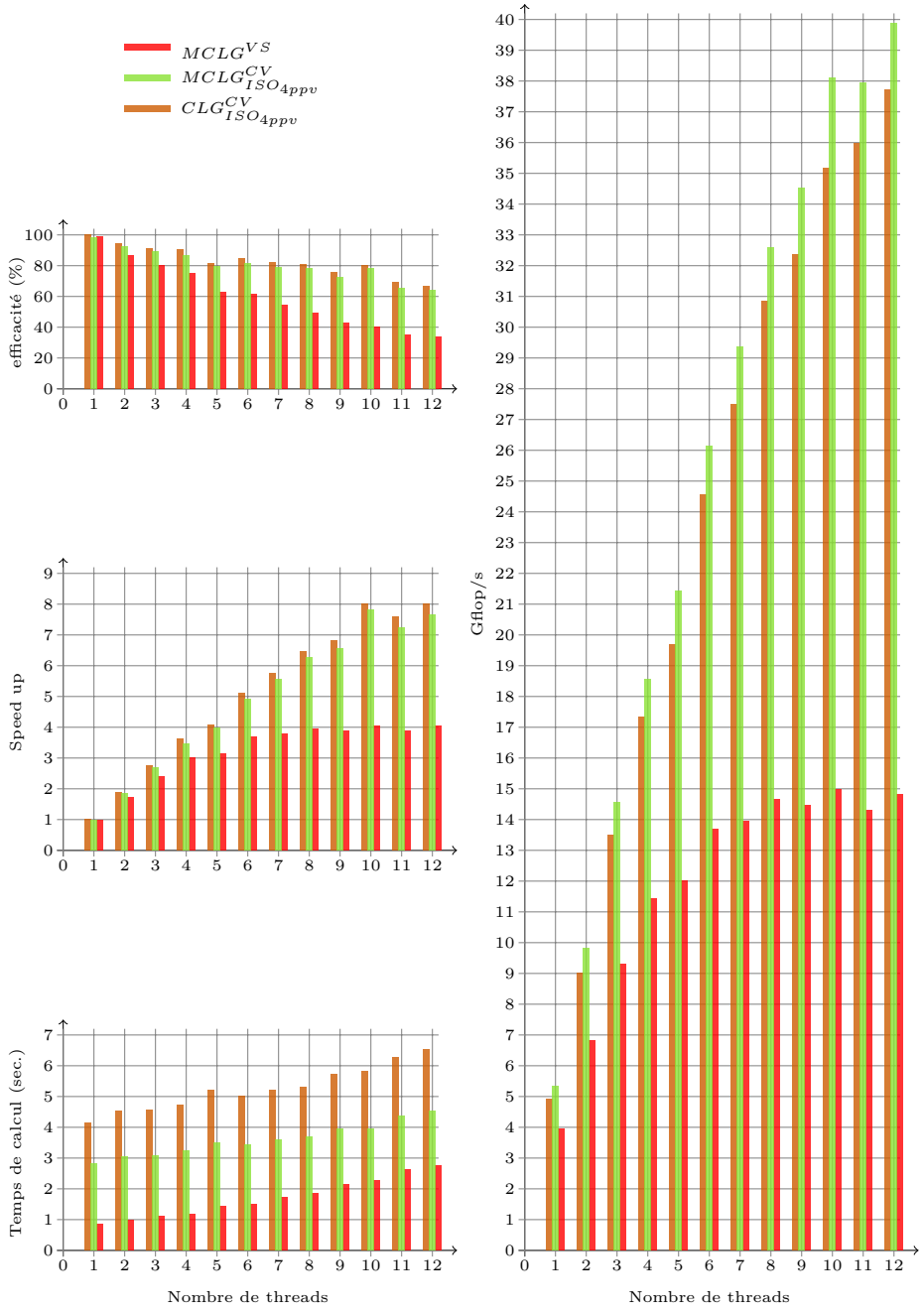

FIGURE 5.6 – Mesure des performances des différentes méthodes développées à charge de calcul fixe par thread (300×300 pixels). **Haut gauche :** efficacité. **Milieu gauche :** accélération. **Bas gauche :** temps de calcul. **droite :** nombre d'opérations flottantes par seconde.

nécessaires à la convergence de la méthode.
Cependant, cet algorithme bénéficie relativement peu de l'augmentation du nombre de cœurs du processeur. Cela se voit dans les deux expériences menées avec nombre de threads variable (*c.f.*FIGURES 5.5 et 5.6) : au delà de 6 threads, dédier davantage de threads aux calculs n'améliore que très peu les performances. Ces résultats encouragent à explorer pour cette méthode une résolution de la partie régularisation par un solveur SOR parallélisé par coloration de mailles pour améliorer l'utilisation du cache mémoire et peut être ainsi améliorer ces performances.
La résolution par calcul variationnel quand à elle se prête bien à la parallélisation. En effet, l'augmentation du nombre de threads permets d'effectuer un nombre d'opérations par seconde de plus en plus grand. L'efficacité des solveurs $MCLG^{CV}_{ISO_{4ppv}}$ et $CLG^{CV}_{ISO_{4ppv}}$ est ainsi au pire des cas de l'ordre de 60% lorsque celle de $MCLG^{VS}$ décroît jusqu'à moins de 40%.
On notera finalement les pic de performances des trois algorithmes étudiés :

	$MCLG^{CV}_{ISO_{4ppv}}$	$CLG^{CV}_{ISO_{4ppv}}$	$MCLG^{VS}$
Gflops/s	43.08	40.66	17.31

5.3 Mise en œuvre GPU/CUDA

L'architecture hautement parallèle des unités de calcul graphique (GPU) offre une puissance de calcul non négligeable. L'apparition des bibliothèques CUDA [Nvidia, 2007] et openCL [Tsuchiyama *et al.*, 2010] a permis une programmation plus aisée des cartes graphiques. Un grand nombre d'applications ont été portées sur cette architecture avec une accélération importante des temps de calcul.
Un examen de l'état de l'art des performances des méthodes actuelles d'estimation du flot optique (sur le site de l'université de Middlebury [1] dédié à l'évaluation des méthodes de flot optique) permet de constater que, parmi celles qui affichent un temps de calcul comparable au notre (les six plus rapides), quatre d'entre elles sont développé sur GPU. Ceci a encouragé le portage des méthodes par calcul variationnel sur ces architectures.
Cette partie commence par une présentation rapide de l'architecture des GPU et des problématiques qui se posent lors de leurs programmations. Quelques mesures de temps seront ensuite présentées pour finir sur une discussion des performances observées.

5.3.1 Le modèle de programmation sur GPU

Le traitement des données sur GPU se fait par instructions simples sur données multiples (SIMD), *i.e.* les mêmes opérations sont effectuées sur un ensemble de données de manière parallèle.
Tous les algorithmes ne se prêtent pas à ce genre de programmation. Les algorithmes présentés dans cette thèse ont été pensés pour permettre un traitement en parallèle.

1. http ://vision.middlebury.edu/flow/eval/

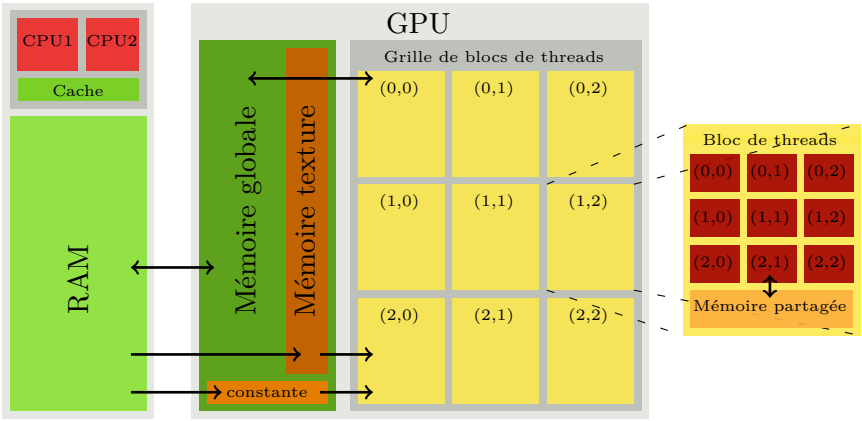

FIGURE 5.7 – Modèle de programmation sur GPU et présentation des différents types de mémoires et des manières d'y accéder.

Leur portage sur GPU est donc possible.
Les unités graphiques sont utilisées comme des co-processeurs pour les CPU. Un programme classique sur GPU se décompose de la manière suivante :

1. transférer des données de la mémoire CPU vers la mémoire GPU,
2. effectuer un ensemble d'opérations sur le GPU,
3. recopier les données traitées depuis la mémoire GPU sur la mémoire CPU.

Les processeurs graphiques se composent d'un grand nombre d'unités de calculs (appelés également "*threads*") capable d'effectuer des opérations simples en parallèle et à haute fréquence. Ces opérations sont regroupées par petites briques de programmes désignées par le terme "kernels".
Les threads sont regroupés par blocs eux-mêmes distribués sur une grille. Cette hiérarchie (représentée dans la FIGURE 5.7) permet d'identifier les threads d'après leurs positions dans le bloc auquel ils appartiennent et les coordonnées de ce dernier dans la grille.

La programmation sur GPU nécessite d'autre part la gestion de la mémoire. En effet, une carte graphique dispose de plusieurs types de mémoire :

la mémoire globale accessible en lecture et écriture par le GPU et le CPU, c'est la plus lente des mémoires disponibles, mais c'est aussi celle qui a la plus grande capacité,

la mémoire texture mémoire rapide, permettant l'interpolation linéaire des données accélérée de manière matérielle. Elle est accessible en lecture seule (voir remarque 5.3.1),

la mémoire partagée mémoire très rapide, accessible en lecture et écriture, mais uniquement aux threads appartenant à un même bloc,

la mémoire constante mémoire très rapide, accessible en lecture seule, de taille réduite,

la mémoire de registre mémoire accessible en lecture et écriture, mémoire la plus rapide mais de taille extrêmement limitée.

L'algorithme $MCLG^{CV}_{ISO_{4ppv}}$ a été porté sur GPU de deux manière différentes : une version utilisant la mémoire globale et l'autre utilisant la mémoire texture.

Remarque 5.3.1 *Il est spécifié dans la documentation fournie par Nvidia [2007] que la mémoire texture 2D n'est accessible qu'en lecture. Au cours de nos expériences nous nous somme rendus compte qu'il est possible d'écrire dans des cases de la mémoire texture où l'on est sûr de ne pas lire lors de l'exécution. C'est exactement la configuration qu'offre la méthode de color labelling employée pour paralléliser les algorithmes de résolution par calcul variationnel. En effet, lors du traitement d'une couleur donnée, on ne lit que les éléments des autres couleurs, qui ne seront pas modifiés à cette étape du traitement (voir la section 3.4).*

5.3.2 Mesure de performances

Pour mesurer les performances des codes GPU, on utilise une carte Nvidia GTX 470 disposant d'une architecture FERMI [Nvidia, 2010] qui représente la technologie la plus évoluée en matière de GPU chez Nvidia.

Le panel d'expériences qui peut être effectué pour évaluer la version GPU de $MCLG^{CV}_{ISO_{4ppv}}$ est assez réduit. En effet, on ne peut pas maîtriser le nombre de threads actifs. Donc les expériences menées à nombre de threads variable dans la section précédente ne sont pas réalisables ici.
La seule mesure de temps possible est en faisant varier la taille des images. On calcule à partir de cela le nombre d'opérations par seconde et l'accélération offerte par le GPU comparée non seulement au code CPU séquentiel (1 thread), mais aussi parallélisé sur 12 cœurs.

Les courbes de la Figure 5.8 représentent cette expérience menée à l'aide des versions du code GPU employant d'une part la mémoire globale et d'autre part la mémoire texture.
Un résultat surprenant de prime abord est le fait que la version du code GPU qui obtient les meilleurs performances est celle qui utilise la mémoire globale (censée être la mémoire la plus lente). Il s'avère cependant que ceci est dû à l'architecture FERMI qui met en place un système de cache pour la gestion de la mémoire globale [Nvidia, 2010]. On utilise ainsi de manière transparente la mémoire partagée, ce qui explique le gain de performances observé. Pour une carte graphique moins évoluée (n'employant pas de système de cache), l'avantage s'inverse et on retrouve de meilleurs performances pour la version du code utilisant la mémoire texture.

Ensuite, en comparant les courbes représentant le nombre d'opérations par seconde, on observe que celle de la version CPU décroît après avoir atteint son maximum pour 8.10^5 pixels alors que la version GPU du code semble ne pas avoir atteint son pic de performance.

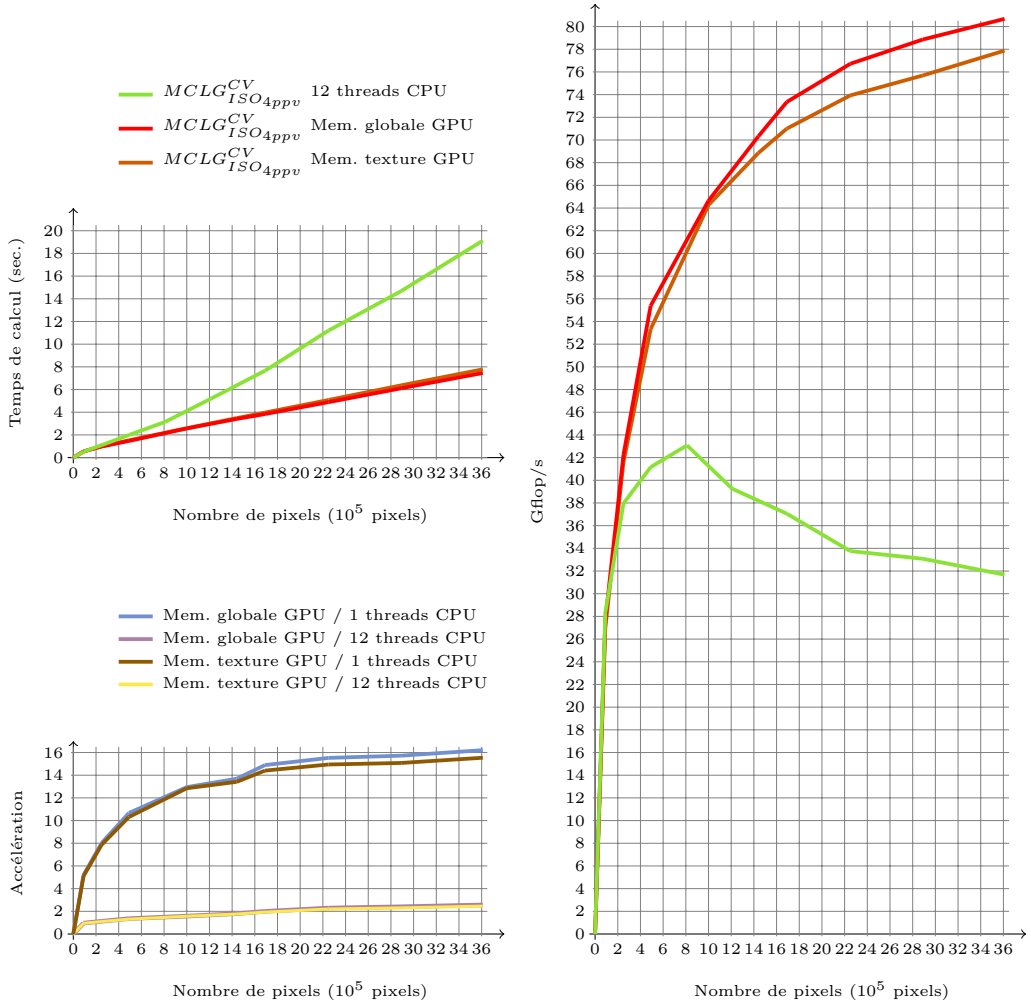

FIGURE 5.8 – Mesure des performances de la version GPU de l'algorithme $MCLG^{CV}_{ISO_{4ppv}}$ comparé à sa version CPU à taille d'images variable (de 300×300 à 1900×1900). Le code GPU emploie soit la mémoire globale soit la mémoire locale. **Haut gauche :** temps de calcul. **Bas gauche :** Accélération. **Droit :** Nombre d'opération par seconde.

Pour ce qui est de l'accélération, l'emploi du GPU permet d'aller jusqu'à 16 fois plus vite que le code CPU séquentiel. Cependant, cette performance est à relativiser, car si l'on compare le code GPU à la version CPU parallèle employant 12 threads, l'accélération n'est plus que de 2.5. Cela reste tout de même une accélération importante.

5.4 Conclusion

Au cours de ce chapitre, nous avons proposé une description détaillée et justifiée de la mise en œuvre des algorithmes développés aux chapitres précédents. Nous avons ensuite effectué une batterie de tests afin de déterminer les caractéristiques du point de vue parallélisme des algorithmes développés sur CPU. Il en est ressorti que la résolution par variable splitting offre sur cette plate-forme les meilleurs temps de calculs.
Cependant, la résolution par calcul variationnel bénéficie plus de l'architecture parallèle car cette dernière offre une plus grande accélération que pour la méthode par variable splitting.
Pour finir, une brève introduction au modèle de programmation sur GPU a été effectuée, puis la mesure des performances a confirmé la puissance de cette architecture comparée à un code CPU séquentiel mais qui doit être relativisée si on la compare à un code CPU parallèle exécuté sur un processeur puissant moderne.

Le chapitre suivant complète la caractérisation des méthodes proposées par l'aspect précision. On utilise pour cela des séquences classiques de traitement d'images, mais aussi des images PIV.

Chapitre 6

Évaluation de la précision des méthodes développées

Sommaire

6.1 Évaluation sur des séquences "classiques" 69

6.2 Évaluation et validation de l'application à la PIV 74

6.2.1 Résolution spatiale 76

6.2.2 Réponse au bruit 78

6.2.3 Réponse fréquentielle 83

6.2.4 Peak locking 86

6.3 Conclusion 86

Ce chapitre se consacre à l'étude de la précision des approches développées aux chapitres 3 et 4. La première partie s'intéressera à l'application de ces méthodes à une batterie de séquences tests disposant d'une solution de référence issue de la base de données de Middlebury [Baker *et al.*, 2009]. Cette première évaluation permettra de mettre en évidence les différentes caractéristiques des solutions des méthodes utilisées. Cette expérience permettra également de se positionner par rapport à l'état de l'art des méthode d'estimation du flot optique.
La seconde partie quant à elle a pour but de valider l'utilisation de la combinaison des méthodes locale et globale à la vélocimétrie par images de particules. On s'appuiera pour cela sur des données synthétiques modélisant des cas particuliers simples permettant de quantifier la résolution spatiale offerte par MCLG ainsi que sa réponse au bruit et sa réponse fréquentielle. Des données expérimentales seront d'autre part utilisées pour observer le phénomène de "peak locking".
Les expériences menées permettront de donner des indications sur la paramétrisation de MCLG lors de sont application à la PIV.

6.1 Évaluation sur des séquences "classiques"

On utilise dans cette section la base de données fournie par l'université de Middlebury [Baker *et al.*, 2009]. Ces séquences sont accompagnées d'un flot optique de référence $\bar{\mathbf{u}} = (\bar{u}, \bar{v}, 1)$, ce qui permet d'évaluer un flot $\mathbf{u}$ mesuré selon différents critères :

– l'erreur angulaire

$$\operatorname{arcos}\left(\frac{\mathbf{u}\cdot\bar{\mathbf{u}}}{\|\mathbf{u}\cdot\bar{\mathbf{u}}\|}\right), \tag{6.1}$$

– l'erreur absolue

$$\|\mathbf{u}-\bar{\mathbf{u}}\|, \tag{6.2}$$

où $\|.\|$ désigne la norme l^2.

Remarque 6.1.1 *Pour éviter les confusions entre le rayon de fenêtre de la méthode locale R qui utilise un noyau uniforme, et celui de MCLG qui utilise un noyau gaussien, on utilisera comme paramètre pour MCLG l'écart type σ de la fenêtre gaussienne (sachant que dans ce cas, on prend un rayon de fenêtre égal à 3σ).*

Dans un premier temps, on présente dans la FIGURE 6.1 les erreurs angulaires obtenues pour les différentes séquences et les différents algorithmes présentés dans cette thèse. On utilise ici pour chaque séquence et pour chaque méthode les paramètres permettant d'obtenir les meilleurs résultats.

Remarque 6.1.2 *Pour obtenir les paramètres optimaux, nous avons choisi l'écart type de la gaussienne σ utilisé pour la convolution parmi $\{0, 0.33, 0.66, 1, 1.33, 1.66\}$ et le paramètre de régularisation α entre $\{10, 20, 30, 40, 50\}$. Dans aucune des séquences étudiées, l'utilisation de la régularisation globale seule (σ=0) n'a permis d'obtenir le meilleur résultat.*

Ce graphique donne une vue d'ensemble des différentes alternatives proposées dans ces travaux. On voit notamment que la résolution par calcul variationnel donne, dans notre cas, de meilleurs résultats que l'approche par séparation de critère.
D'autre part, on remarque que l'approximation isotrope de la régularisation par variation totale (3.3) produit toujours une erreur plus faible que celle anisotrope (3.6).
Dans ce cadre, on note également que le traitement de la partie régularisation ISO_{8ppv} permet en général d'obtenir des erreurs toujours inférieures ou égale à celles de ISO_{4ppv}. La FIGURE 6.2 permet d'illustre ce fait en représentant les solutions de la séquence Urban3 pour lesquelles la différence entre ces deux méthodes est la plus grande. Le flot optique fourni par la méthode ISO_{8ppv} reproduit des détails plus fins ainsi que des bords plus nets par rapport à celui de ISO_{4ppv}.

La FIGURE 6.1 permet également de comparer la méthode de warping utilisée dans CLG par Bruhn *et al.* et la modification proposée au chapitre 2 dans MCLG. On remarque que pour la majorité des séquences étudiées, MCLG fournit de meilleurs résultats que CLG. Pour confirmer cette observation, on évalue les méthodes $CLG^{CV}_{ISO_{8ppv}}$ et $MCLG^{CV}_{ISO_{8ppv}}$ selon la méthodologie proposée par Middlebury, *c. à d.* des paramètres fixes seront utilisés pour toutes les séquences :

– CLG : $\sigma = 1.33$ et $\alpha = 10$,

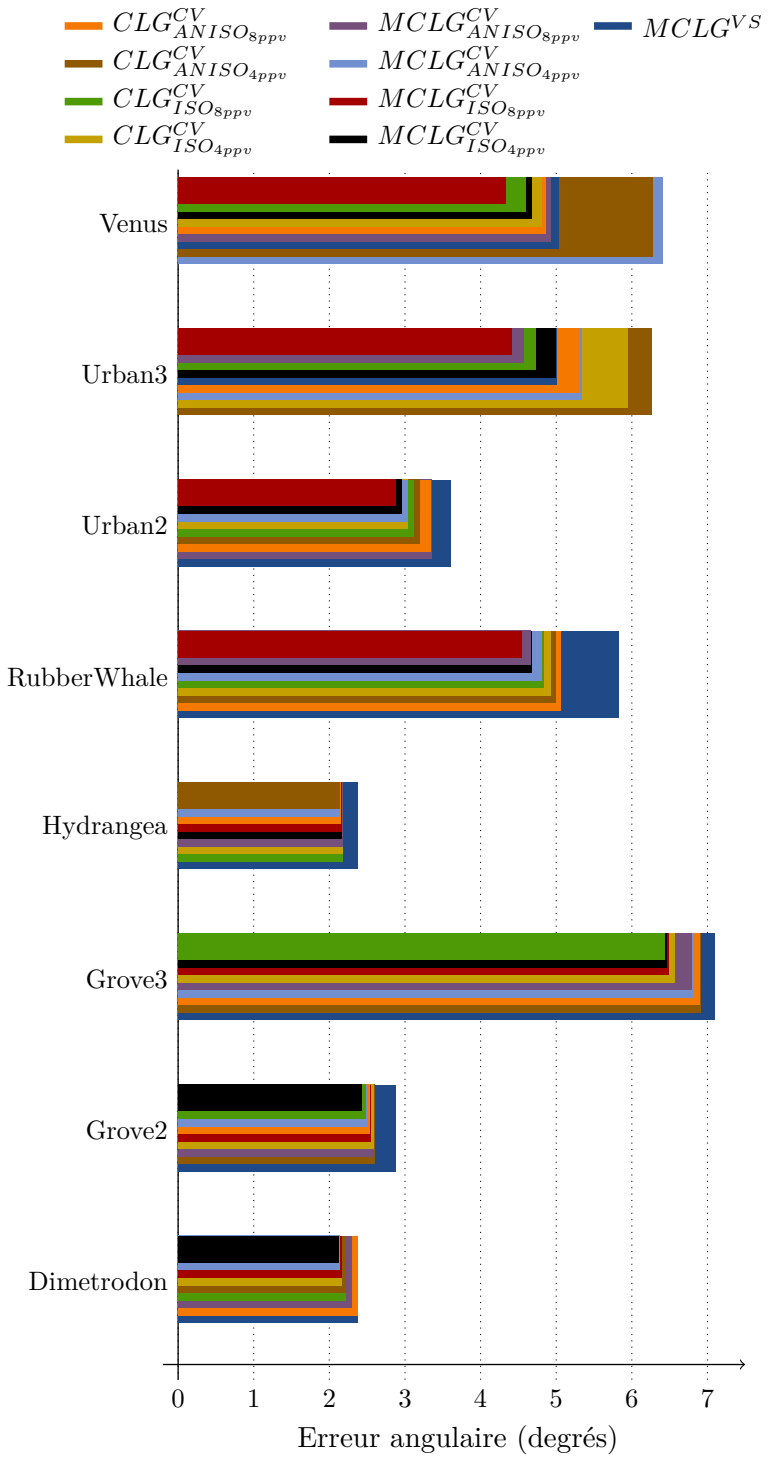

FIGURE 6.1 – Compilation des meilleurs résultats en terme d'erreur angulaire pour les sept algorithmes développés au cours de cette thèse sur les données de test de Middlebury.

FIGURE 6.2 – Comparaison des méthodes de gestion de la partie régularisation sur le cas Urban3. **À gauche :** meilleur solution pour $MCLG^{CV}_{ISO_{8ppv}}$ (α=10, $\sigma = 0.66$). **À droite :** meilleur solution pour $MCLG^{CV}_{ISO_{4ppv}}$ (α=10, σ=1).

– MCLG : $\sigma = 0.66$ et $\alpha = 20$.

Ces paramètres ont été sélectionnés comme étant le couple (σ, α) parmi ceux spécifiés dans la remarque 6.1.2 qui permettent d'obtenir l'erreur moyenne la plus faible sur l'ensemble des séquences.

Le tableau 6.1 liste la moyenne des erreurs angulaires et absolues mesurées pour toutes les séquences sous les conditions précédemment citées. MCLG prend l'avantage sur CLG dans la majorité des cas. Cependant, le cas Urban3 est intéressant, en effet CLG bât largement MCLG du point de vue de l'erreur angulaire, pour laquelle la différence atteint 1.23 degrés, mais cet écart est à relativiser à la vue de la différence de leurs erreurs absolues moyennes qui est de 5 centièmes de pixel.

Tableau 6.1 – Comparaison des méthodes CLG ($\alpha = 10, \sigma = 1.33$) et MCLG ($\alpha = 20, \sigma = 0.66$) selon le principe d'évaluation de Middlebury : paramètres fixés pour toutes les séquences.

	Erreur angulaire moyenne		Erreur absolue moyenne	
	CLG	MCLG	CLG	MCLG
Dimetrodon	**2.21**	2.33	**0.11**	0.12
Grove2	2.61	**2.59**	0.18	0.18
Grove3	**6.53**	6.54	0.70	**0.68**
Hydrangea	2.30	**2.20**	0.19	**0.18**
RubberWhale	4.84	**4.54**	0.15	**0.14**
Urban2	3.11	**2.87**	0.50	**0.39**
Urban3	**4.73**	6.00	**0.61**	0.66
Venus	4.89	**4.78**	0.32	**0.30**

Les résultats correspondant aux mesures d'erreurs du tableau 6.1 sont présentés dans la FIGURE 6.3. On y montre de plus les résultats de la méthode $MCLG^{VS}$.

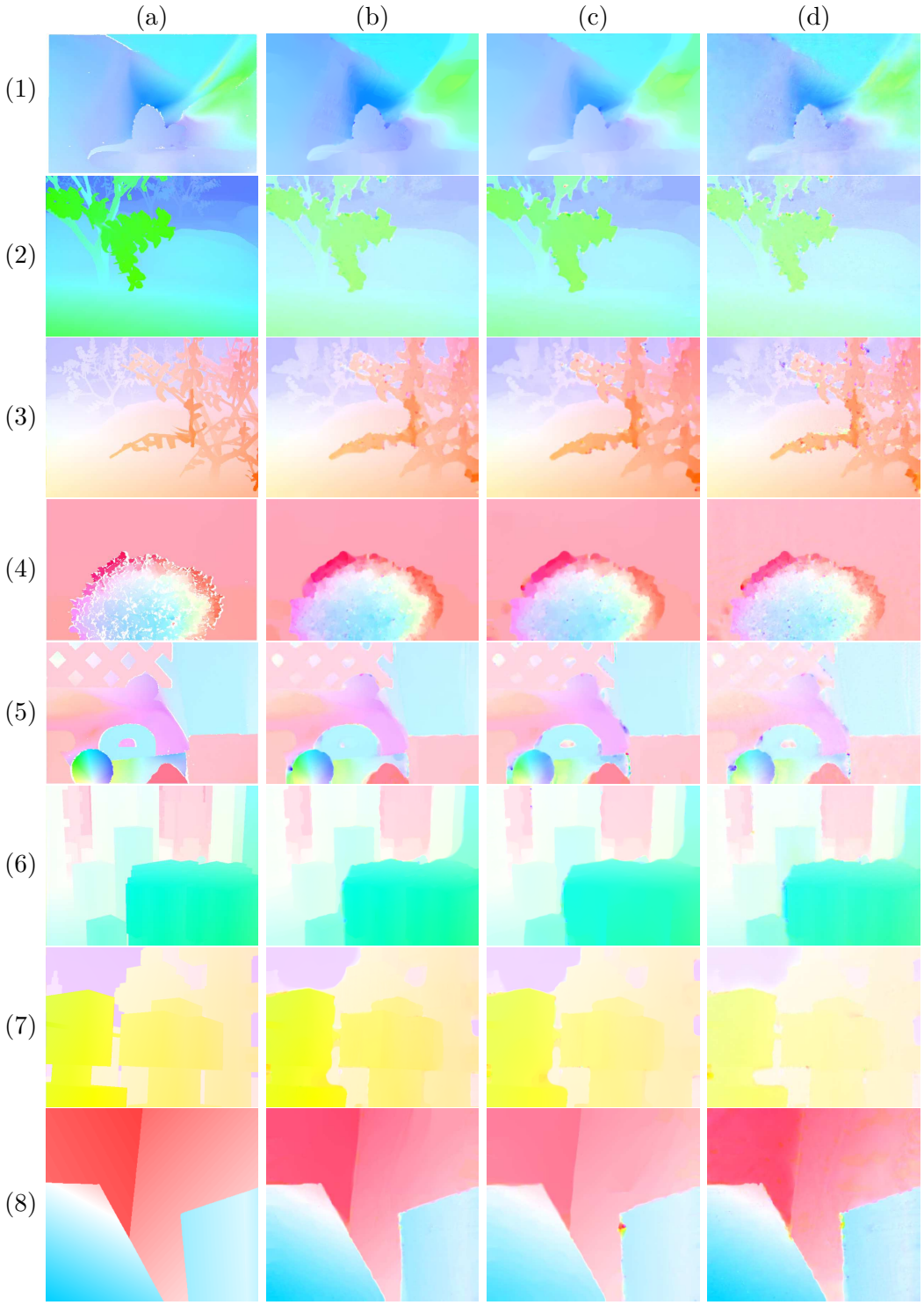

FIGURE 6.3 – Résultats sur les quels les erreurs du tableau 6.1 sont mesurées. Ligne : **(1)** Dimetrodon, **(2)** Grove2, **(3)** Grove3, **(4)** Hydrangea, **(5)** RubberWhale, **(6)** Urban2, **(7)** Urban3, **(8)** Venus. Colonne : **(a)** Solution de référence. **(b)** $MCLG^{CV}_{8ppv_{ISO}}$. **(c)** $CLG^{CV}_{8ppv_{ISO}}$. **(c)** $MCLG^{VS}$.

6.2 Évaluation et validation de l'application à la PIV

On utilise des images aussi bien synthétiques qu'expérimentales dans les évaluations menées dans la suite. On étudiera différentes caractéristiques de MCLG comparées à celles des méthodes locale et globale seules en utilisant différents paramètres.

Des mesures d'erreurs sont d'abord effectuées pour la séquence présentée dans la FIGURE 2.1 pour quantifier l'amélioration offerte par la combinaison des méthodes locale et globale observé au chapitre 2. On présente dans la FIGURE 6.4 la compilation des meilleurs résultats (en terme d'erreur angulaire) obtenus sur cette séquence à l'aide de l'ensemble des algorithmes développés dans cette thèse.
On remarque tout d'abord que la combinaison des méthodes locale et globale permet d'améliorer les résultats de ces deux approches utilisées séparément. On voit également que la correction apportée à la partie adéquation aux données dans MCLG produit des erreurs plus faibles que dans la version originale de CLG. D'autre part, la résolution par calcul variationnel produit de meilleurs résultats que celle par séparation de critère. Dans ce cadre, la régularisation quadratique (utilisée pour produire les résultats du chapitre 2) donne des erreurs plus importantes que celle de la régularisation par variation totale.
Cependant, la manière de linéariser la partie régularisation par variation totale dans le cadre de la résolution par calcul variationnel n'influence que très peu la qualité de la solution. En effet, la capacité des approches développées dans cette thèse (ISO_{8ppv} et $ANISO_{8ppv}$) à mieux restituer les discontinuités du flot (voir FIGURE 6.2) n'est pas exploitée dans ce cas.

On décide à partir de maintenant d'utiliser uniquement la méthode $MCLG^{CV}_{ISO_{4ppv}}$ dans les expériences qui vont suivre.

Le cas étudié ensuite (*c.f.*FIGURE 6.5) représente une variation brusque de 0.25 à 0.35 pixel dans un déplacement horizontal. Cette expérience nous a été suggérée par B. Wieneke. Ce cas simple va permettre de visualiser la résolution spatiale ainsi que la robustesse au bruit des méthodes testées.

Remarque 6.2.1 *Les déplacements dans ces séquences étant d'intensité réduite, on n'utilisera qu'un seul niveau de grille et trois relinéarisations (warp) du système lors de la résolution.*

On présente dans un premier temps le problème que pose le choix des paramètres utilisés dans les méthodes locale et globale dans la FIGURE 6.6. Un rayon de fenêtre trop grand pour la méthode locale implique une mauvaise restitution de la solution au niveau de la transition (zone encadrée en orange) alors qu'un rayon de fenêtre trop faible produit des perturbations dans les zones homogènes de la solution (Zone encadrée en vert).

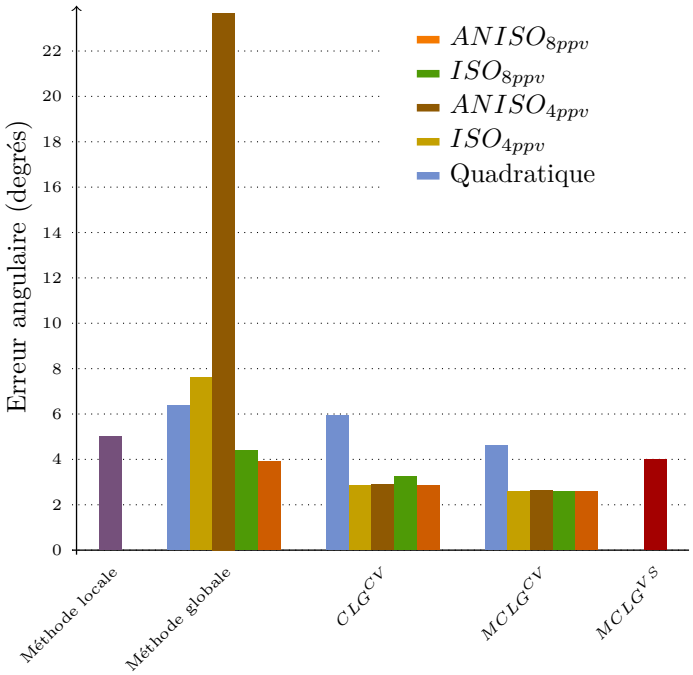

FIGURE 6.4 – Compilation des meilleurs résultats en terme d'erreur angulaire pour les différents algorithmes développés au cours de cette thèse sur le cas synthétique d'images de particules représentant une grille de tourbillons (*c.f.*FIGURE 2.1).

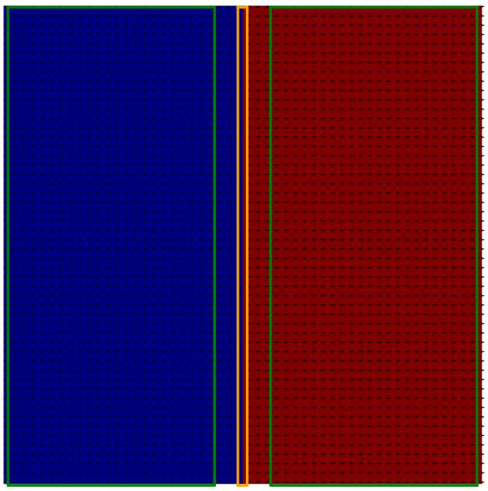

FIGURE 6.5 – Norme du champ de vecteurs présentant une variation horizontale brusque de 0.25 à 0.35 pixel. Les zones encadrées sont celles étudiées dans la suite. **En orange :** zone de transition qui servira à évaluer la résolution. **En vert :** mouvement vertical uniforme, à gauche, déplacement de 0.25 pixel, à droite, déplacement de 0.35 pixel. On évaluera sur cette zone la réponse au bruit.

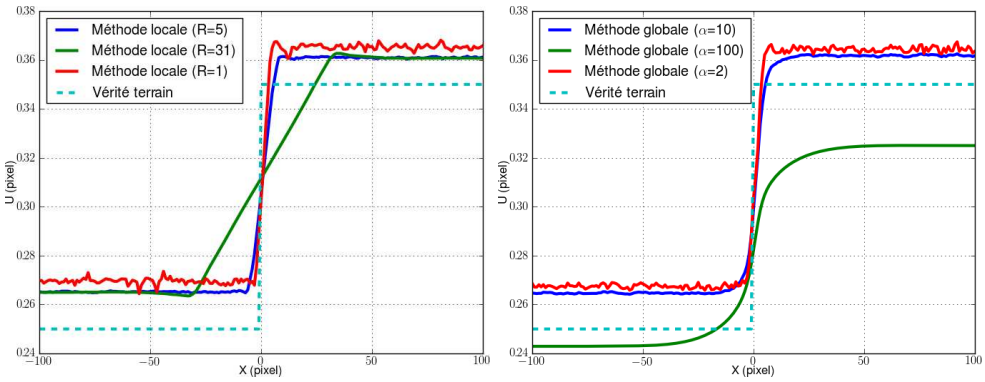

FIGURE 6.6 – Profil moyen du déplacement horizontal pour le cas de la FIGURE 6.5. **Gauche :** Les solutions de la méthode par régularisation locale avec un rayon d efenêtre R=1, 5 et 31. **Droite :** Les solutions de la méthode par régularisation globale avec paramètre de régularisation α=2, 10 et 100.

Pour le cas de l'approche globale, un paramètre de régularisation α trop important entraîne une sous estimation de l'amplitude du mouvement que l'on ne retrouve pas avec la méthode locale. Si en revanche α est choisi trop faible, la solution se dégrade dans les zones homogènes.

Cet exemple simple permet d'évaluer la capacité des algorithmes à restituer les détails du mouvement (au niveau de la variation brusque) ainsi que leurs sensibilités au bruit (dans les zones homogènes).

6.2.1 Résolution spatiale

La zone étudiée ici est celle encadrée en orange dans la FIGURE 6.5, *c. à d.* au niveau de la variation brusque. La résolution est représentée dans nos expériences par l'inverse de la pente de la courbe au niveau de la discontinuité du flot normalisée par l'amplitude de la variation. Donc plus importante est la résolution d'un résultat, moins brusque est la transition entre les zones de déplacement de 0.25 et 0.35 pixel.

La FIGURE 6.7 présente la comparaison des résolutions des méthodes locale et globale en fonction de leurs paramètres respectifs R et α. Comme on pouvait le prédire la résolution de la méthode locale est équivalente au diamètre de la fenêtre utilisée. On voit également que la résolution augmente logiquement en fonction de α.

On présente dans la FIGURE 6.8 la résolution de MCLG avec α variable pour différents σ fixés. Il est clair que plus σ augmente, moins la résolution varie en fonction de α. Ceci nous permet de conclure que pour cette méthode, la résolution est régie par σ donc par la partie régularisation locale du modèle.

Pour finir, la FIGURE 6.9 montre les paramètres des différentes méthodes (locale,

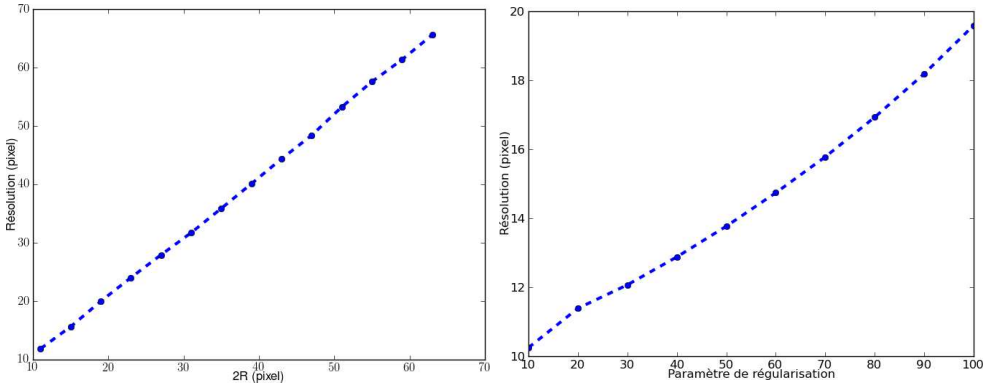

FIGURE 6.7 – **À gauche :** Résolution spatiale de la méthode par régularisation locale en fonction du diamètre de la fenêtre. **À droite :** Résolution spatiale de de la méthode globale en fonction du paramètre de régularisation.

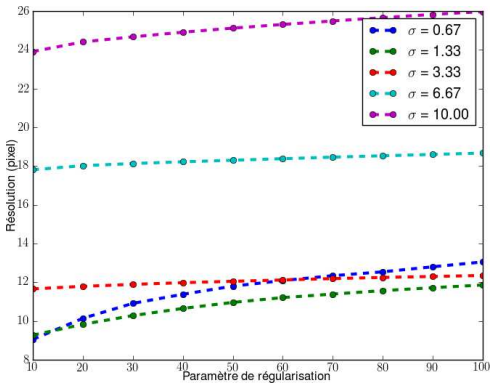

FIGURE 6.8 – Résolution spatiale de la méthode MCLG avec différents rayons de fenêtre d'interrogation en fonction du paramètre de régularisation α.

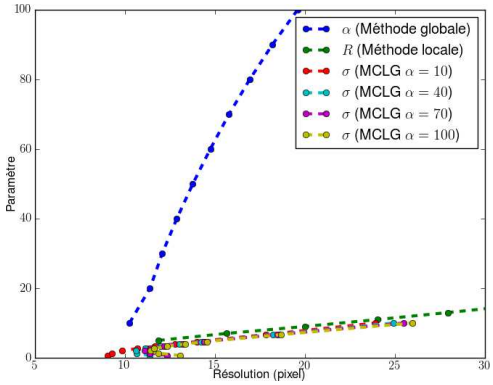

FIGURE 6.9 – Les différents paramètres des méthodes locale, globale et MCLG en fonction de la résolution mesurée. Ces courbes permettent d'identifier l'influence de ces paramètres sur la résolution.

globale et MCLG) en fonction de la résolution mesurée. Ces courbes permettent de visualiser l'influence des différents paramètres et confirment que la résolution de la méthode MCLG dépends essentiellement de σ qui suit une courbe proche du rayon de fenêtre de la méthode locale.

En conclusion, la meilleure résolution est obtenue pour la méthode locale avec un rayon de fenêtre faible (R=5), pour la méthode locale avec un paramètre de régularisation faible ($\alpha = 10$) et pour MCLG avec un couple (α, σ) de faible valeurs (respectivement 10 et 4).

Les mesures effectuées ici vont nous permettre dans les expériences suivantes de s'affranchir des différents paramètres lors de la comparaison des différentes méthodes. On utilisera en effet la résolution qui leur correspond.
L'étude des différentes méthodes se poursuit dans la section suivante pour évaluer leur comportement en présence de différents types de bruits.

6.2.2 Réponse au bruit

Pour représenter la précision des méthodes étudiées et leur robustesse au bruit, la séquence représentée par la FIGURE 6.5 sera générée en supposant quatre cas correspondant à différentes conditions expérimentale (voir section 1.4.2) :

cas 0 pas de bruit ajouté (correspond aux images utilisée dans la section précédente pour évaluer la résolution des méthodes),

cas 1 ajout de mouvement hors plan, de sorte que certaines particules présentes dans la première image peuvent disparaître dans la seconde,

cas 2 ajout d'un bruit gaussien CCD d'écart type de vingt niveaux de gris,

cas 3 réduction de la densité de particules additionné au même bruit du cas 2.

L'expérience consiste à mesurer l'erreur absolue moyenne (EAM) de la composante horizontale du mouvement sur la zone délimitée par les lignes vertes dans

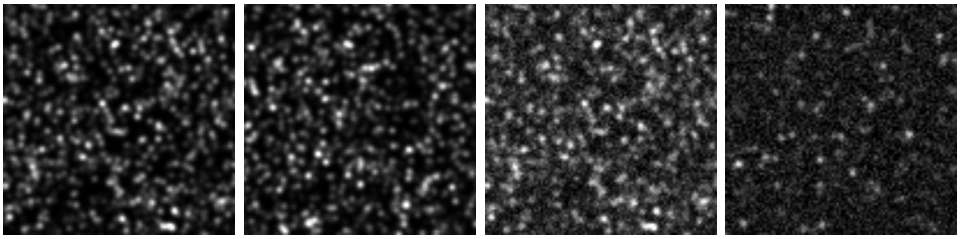

FIGURE 6.10 – Agrandissement d'une même zone d'une image de la séquence FIGURE 6.5 dans les différentes conditions considérées. **De gauche à droite :** cas 0,1,2 et 3.

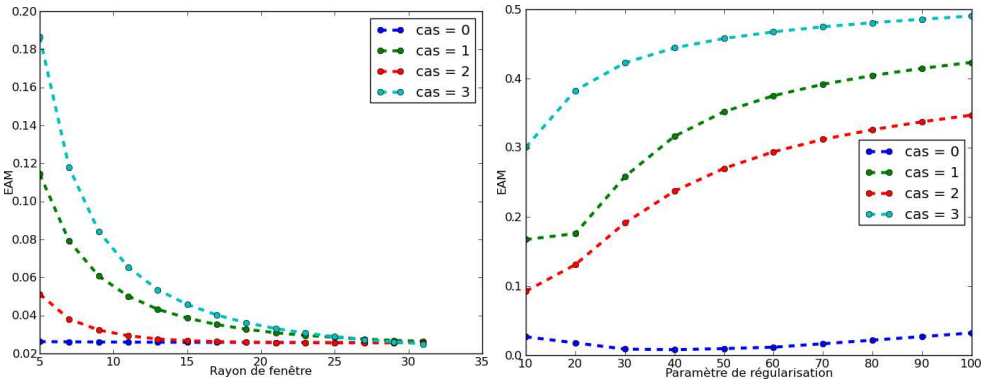

FIGURE 6.11 – **À gauche :** Erreurs absolues moyennes mesurées en fonction du rayon de la fenêtre d'interrogation de la méthode locale sur les quatre cas étudiés. **À droite :** Erreurs absolues moyennes mesurées en fonction du paramètre de régularisation α de la méthode globale sur les quatre cas étudiés.

la FIGURE 6.5.

On commence par mesurer les performances de la méthode par régularisation locale et globale dans la FIGURE 6.11. Ces courbes confirment la robustesse au bruit de la première approche. En effet, l'utilisation d'un rayon de fenêtre assez grand garantit une erreur faible dans tout les cas étudiés. En revanche, si la méthode globale se comporte bien lorsqu'il n'y a pas de bruit (cas 0), elle diverge jusqu'à obtenir une erreur moyenne d'un demi pixel dans le cas 3, ce qui est assez élevé car le déplacement varie de 0.25 à 0.35.

On représente maintenant dans la FIGURE 6.12 les performances de la méthode MCLG. On notera que le fait de combiner les méthodes locale et globale corrige cette sensibilité au bruit. Ainsi, les erreurs des solutions de la méthode MCLG sont comparables, dans les quatre cas, à celles mesurées dans le cas de la méthode locale seule (*c.f.*FIGURE 6.11). Il faut cependant choisir un écart type σ assez grand (supérieur à 1.33) pour garantir un niveau d'erreur faible indépendamment du paramètre de régularisation α.

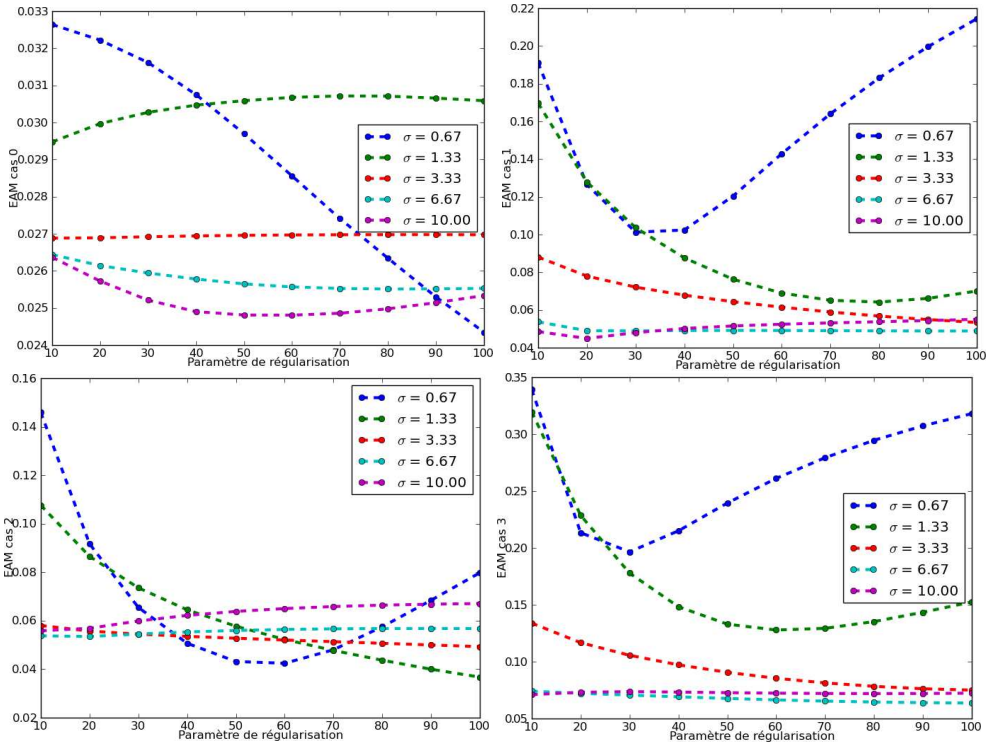

FIGURE 6.12 – Erreurs absolues moyennes mesurées en fonction du paramètre de régularisation de MCLG avec différents écarts types de fenêtre gaussienne σ. **Haut gauche :** cas 0. **Haut droit :** cas 1. **Bas gauche :** cas 2. **Bas droit :** cas 3.

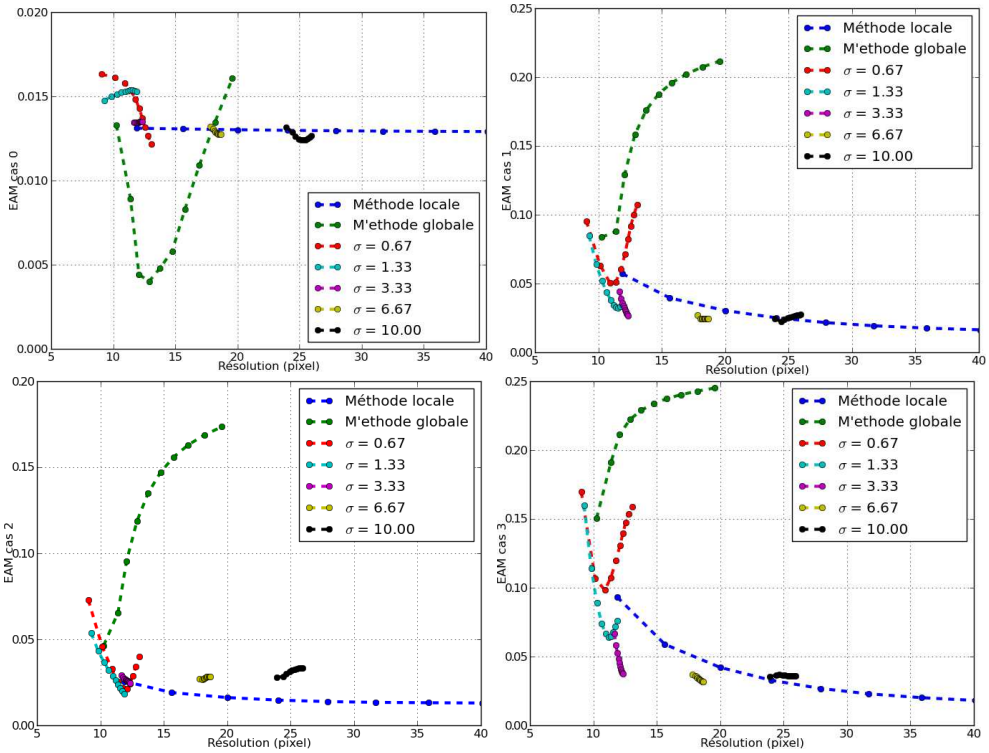

FIGURE 6.13 – Erreurs absolues moyennes mesurées en fonction de la résolution des méthodes locale, globale et MCLG avec différents paramètres. **Haut gauche :** cas 0. **Haut droit :** cas 1. **Bas gauche :** cas 2. **Bas droit :** cas 3.

Pour mettre en valeur les résultats de MCLG comparés aux régularisations locale et globale seules, les EAM mesurées précédemment sont représentées en fonction des résolutions respectives des différentes méthodes. En effet, une solution idéale présente une résolution basse et une erreur absolue moyenne faible.
Donc, dans les courbes présentées dans la FIGURE 6.13, plus une mesure est proche de l'origine du repère, meilleur est son rapport résolution/bruit. Ces résultats confirment le fait que la méthode globale offre une meilleure résolution que la méthode locale au pris d'une grande sensibilité au bruit.
MCLG offre le compromis espéré en alliant une meilleure résolution ainsi qu'une robustesse au bruit comparable à celle de la méthode locale.

Pour finir cette partie, on présente dans la FIGURE 6.14 le profil moyen de la solution qui offre le meilleur compromis résolution/bruit pour les méthodes locale et globale ainsi que le profil de la solution de MCLG qui offre le même niveau d'EAM que la méthode locale. On observe dans tout les cas, d'une part, le meilleur comportement de la méthode MCLG comparé à l'approche par régularisation locale au niveau d ela rupture, et d'autre part, la correction du biais présent dans la solution de la méthode globale dans les différents cas bruités.

Remarque 6.2.2 *Les écarts entre les courbes présentées dans la* FIGURE *6.14 peuvent paraître importants, mais ils sont seulement de l'ordre de quelques centièmes*

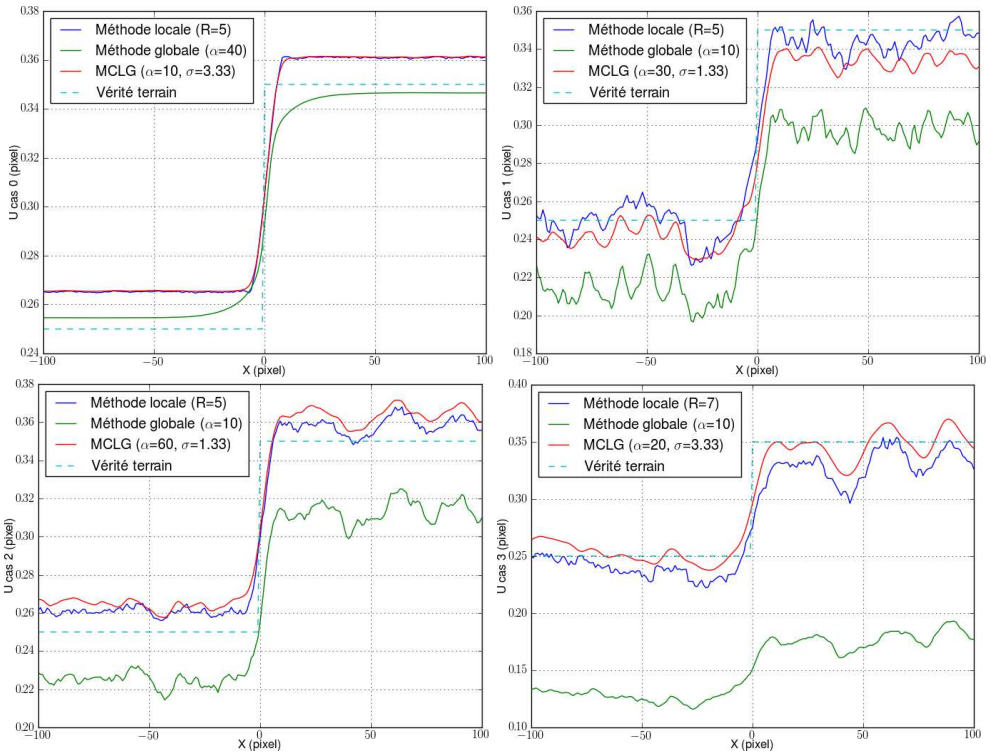

FIGURE 6.14 – Comparaison des profils moyens des solutions de MCLG, méthode locale et méthode globale pour le meilleur compromis bruit/résolution. **Haut gauche :** cas 0. **Haut droit :** cas 1. **Bas gauche :** cas 2. **Bas droit :** cas 3.

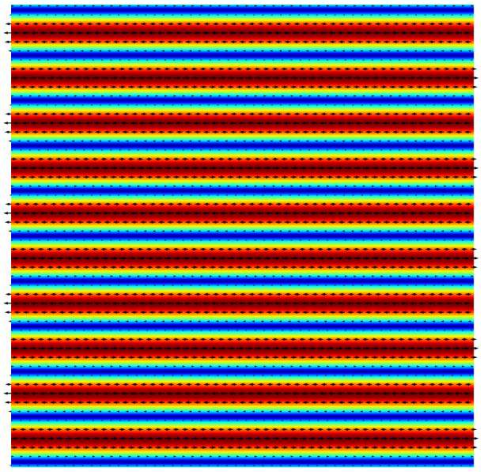

FIGURE 6.15 – Norme du champ de vecteurs représentant un cisaillement sinusoïdal d'amplitude 2 pixels et de longueur d'onde 200 pixels.

de pixel à l'exception du cas 3 pour lequel la méthode globale ne converge pas.

6.2.3 Réponse fréquentielle

Pour évaluer la réponse fréquentielle d'une méthode, Scarano et Riethmuller [2000] proposent de la tester sur une séquence représentant un cisaillement sinusoïdal avec plusieurs longueurs d'onde :

$$(u(x,y), v(x,y)) = \left(A \sin\left(2\pi \frac{y}{l}\right), 0\right) \tag{6.3}$$

où $u(x,y)$ et $v(x,y)$ sont respectivement les composantes horizontale et verticale du champ de vecteur aux coordonnées (x,y). Dans les séquences considérées, l'amplitude A_{true} est de 2 pixels et la longueur d'onde l varie de 20 jusqu'à 400. La FIGURE 6.15 représente la norme du déplacement pour le cas $l = 200$. Comme pour le cas précédent, les déplacements sont de faible amplitude, on conservera donc les même paramètres pour le solveur, à savoir, un seul niveau de grille et 3 re-linéarisations du système.

La FIGURE 6.16 illustre encore une fois le compromis que représente le choix des paramètres utilisés dans les méthodes locale et globale : un rayon de fenêtre R ou un paramètre de régularisation α trop important entraîne une sous estimation des amplitudes du déplacement. Cela pose notamment problème sur les données bruitées qui nécessitent une forte régularisation.

Remarque 6.2.3 *Le fameux effet de la régularisation par variation totale qui favorise les fonctions constantes par morceaux est bien visible dans le cas de la méthode par régularisation globale avec α=90 dans la* FIGURE *6.16.*

La réponse fréquentielle d'une méthode est obtenue en représentant le rapport $\frac{A}{A_{true}}$ (entre l'amplitude mesurée et l'amplitude réelle) en fonction de la résolution

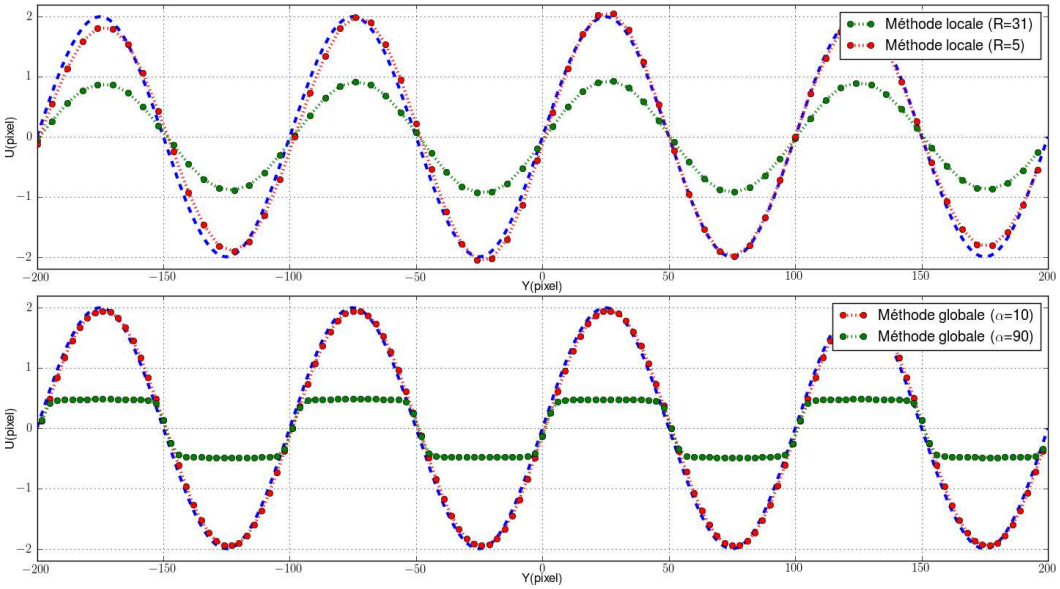

FIGURE 6.16 – Profil moyen du déplacement horizontal pour le le cas du cisaillement sinusoïdal de longueur d'onde 100 pixels. **En haut :** Les solutions de la méthode par régularisation locale avec un rayon R=5 et 31. **En bas :** Les solutions de la méthode par régularisation globale avec paramètre de régularisation α=10 et 90.

normalisée par rapport à la longueur d'onde l pour la méthode locale. Les méthodes étudiées sont comparées à une moyenne glissante dont la réponse fréquentielle théorique est représentée en pointillés noirs dans les FIGURES 6.17 et 6.18.

Les réponses fréquentielles des méthodes locale et globale sont tracées dans la FIGURE 6.17. On observe que la courbe de la méthode locale suit celle de la moyenne glissante. Pour sa part, la méthode globale est très sensible à l'augmentation de la régularisation qui entraîne une sous-estimation importante du mouvement pour les grandes valeurs de α.

L'étude de la méthode MCLG est quant à elle effectuée à la FIGURE 6.18. On y représente la réponse fréquentielle avec différents écarts types de fenêtre gaussienne σ pour des paramètres de régularisation α fixés. Comme déjà observé dans le cas de la variation brusque (*c.f.*FIGURE 6.5) précédemment étudié, la méthode MCLG est plus sensible aux variations de σ qu'à celles de α.
En effet, plus on augmente l'écart type du noyau gaussien σ, moins la variation de l'amplitude semble dépendre du paramètre de régularisation.

Pour finir l'étude de ces séquences, on présente dans la FIGURE 6.19 les meilleurs résultats de MCLG et de la méthode locale obtenus pour différentes longueurs d'ondes. On ne représente pas la meilleure solution de la méthode globale car elle suit sensiblement celle de MCLG.
On observe dans un premier temps que pour le cas des longueurs d'onde 60 et 100 MCLG suit plus fidèlement la courbe réelle. Ensuite, il est important de noter que

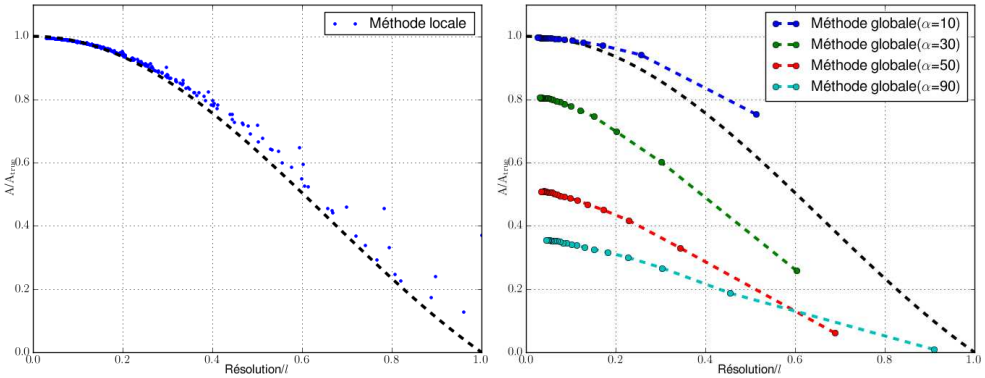

FIGURE 6.17 – Ratio entre l'amplitude de la solution et l'amplitude réelle du mouvement en fonction de la résolution normalisée. **À gauche :** méthode locale avec différents rayons de fenètre. **À droite :** méthode globale en fonction de différents paramètres de régularisation.

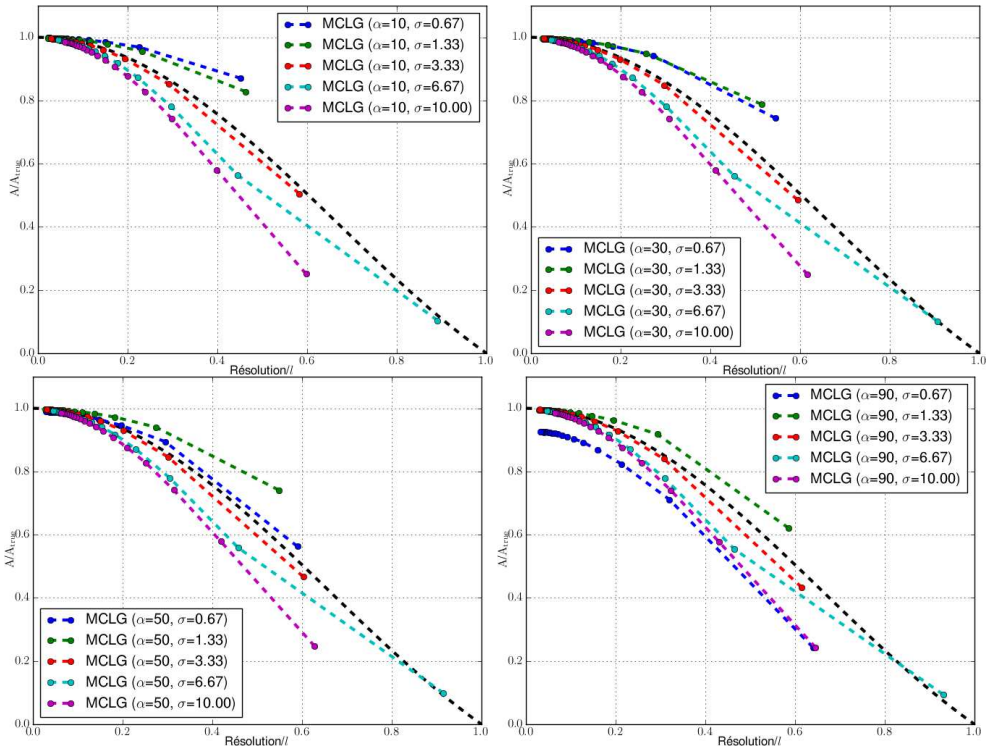

FIGURE 6.18 – Ratio entre l'amplitude de la solution de MCLG et l'amplitude réelle du mouvement en fonction de la résolution normalisée. Les résultats de différents écarts types de fenêtre gaussienne σ sont présentés avec des paramètres de régularisation α fixés.

pour éviter la surestimation de l'intensité du mouvement et ainsi obtenir le meilleur résultat de la méthode locale, il est nécessaire d'adapter la taille du rayon de fenêtre utilisé en fonction de la longueur d'onde du signal. En revanche, les paramètres utilisés pour MCLG sont les même pour toutes les longueurs d'onde.
C'est un grand avantage car dans les cas réels plus complexes, on rencontre différentes longueurs d'ondes, et MCLG permet de toutes correctement les appréhender avec des paramètres fixés. Dans les même conditions, une méthode locale détériorerait un certain nombre de longueurs d'ondes pour lesquelles le rayon de fenêtre utilisé n'est pas adapté.

6.2.4 Peak locking

On utilise dans cette partie des données réelles tirées du PIV challenge 2003 [Stanislas *et al.*, 2005]. Cette séquence composée de 100 paires d'images représente un jet turbulent (voir FIGURE 1.5). On présente dans la FIGURE 6.20 un exemple de flot optique instantané (celui de la première paire d'images) accompagné du champ de vecteurs moyen correspondant à l'ensemble de la séquence.

Les images étudiées comportent un certain nombre de difficultés, notamment des particules assez fines pour entraîner un effet de "peak locking" (voir la section 1.4). Pour observer le degrés de sensibilité de $MCLG^{CV}_{ISO_{4ppv}}$ à ce phénomène, on représente dans la FIGURE 6.21 l'histogramme de la composante horizontale ainsi que celui des déplacements fractionnaires pour l'ensemble des flots optiques instantanés calculés.
On remarque bien l'apparition de pics au niveau des valeurs entières dans le premier histogramme mais à un degrés comparable à celui des meilleurs logiciels actuels de traitement d'images PIV (voir [Stanislas *et al.*, 2005]).
On retrouve également une manifestation du peak locking dans l'histogramme des déplacements fractionnaires pour lequel le maximum est atteint en 0. Cet histogramme reste tout de même assez plat, preuve d'un niveau de peak locking peu prononcé.

6.3 Conclusion

Les expériences menées au cours de ce chapitre ont permis d'exhiber certaines caractéristiques de la méthode combinant les approches locales et globales. En ce qui concerne l'étude de cas d'images "classiques", nous avons pu constater le bénéfice de l'emploi de la correction apportée par MCLG à la mise en œuvre du warping dans CLG. La preuve de l'efficacité de la gestion de la partie régularisation développée dans ces travaux a également été apportée, plus particulièrement la version isotrope (ISO_{16ppv}) qui permet d'obtenir les meilleurs résultats sur les séquence de Middlebury.

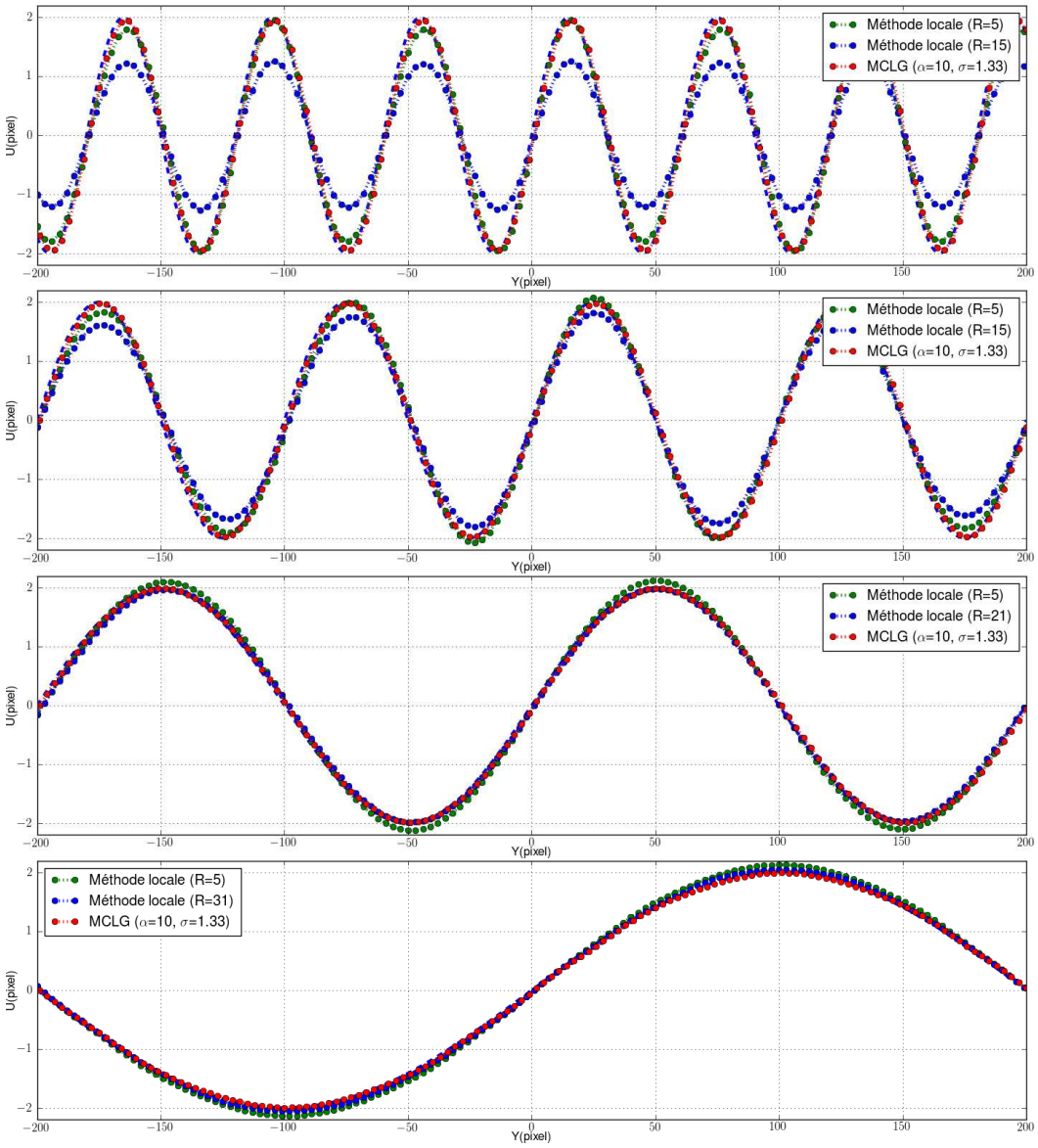

FIGURE 6.19 – Séquence représentant le cisaillement sinusoïdal : profil moyen de la solution de la méthode par régularisation locale comparé à celui de la méthode MCLG pour différentes longueurs d'ondes (de haut en bas, l= 60, 100, 200, 400).

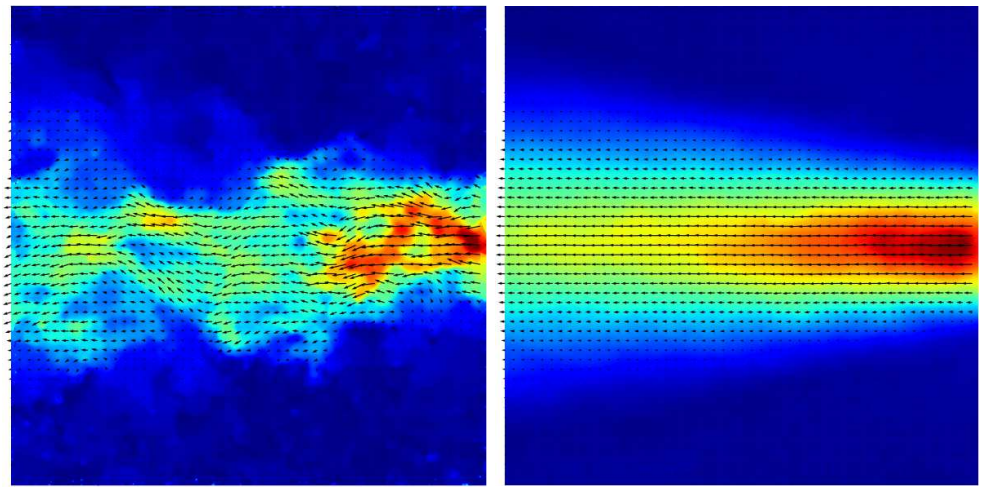

FIGURE 6.20 – Flot optique calculé à l'aide de MCLG de la séquence A du PIV challenge 2003 (*c.f.*FIGURE 1.5) ainsi que sa norme. **À gauche :** Déplacement instantané correspondant à la première paire d'images. **À droite :** Déplacement moyen mesuré sur les 100 paires d'images.

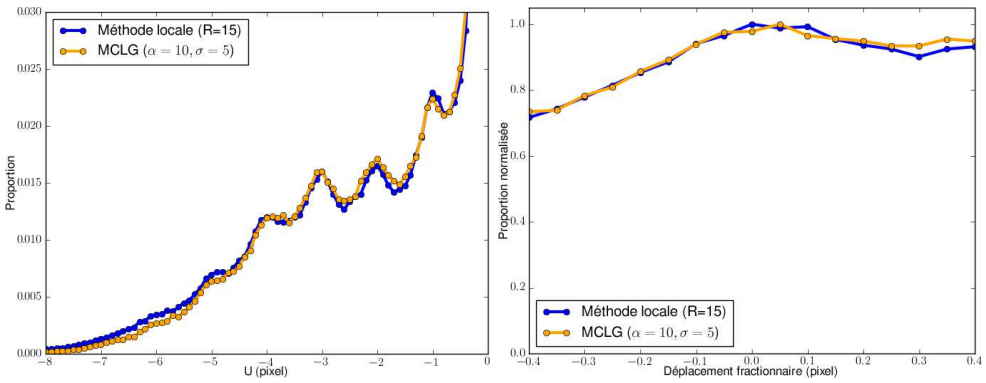

FIGURE 6.21 – On cherche à observer le peak locking sur l'ensemble de la séquence A du PIV challenge 2001. **À gauche :** Histogramme du déplacement horizontal. **À droite :** Histogramme des déplacements fractionnaires.

Pour le cas particulier du traitement d'images de PIV, les résultats obtenus ont permis de montrer que l'objectif d'allier les avantages des méthodes locales et globales a été atteint en combinant ces deux approches. En effet, la méthode MCLG offre la précision des méthodes globales assortie de la robustesse au bruit des méthodes locales. Ces caractéristiques ont été établies en évaluant la résolution (définie comme le nombre de pixels nécessaires pour franchir une rampe) des méthodes étudiées et leurs réponses à différents cas de bruit.
Les expériences menées ont également prouvé la capacité de la méthode MCLG à correctement restituer un large panel de fréquences de signal avec un jeu de paramètres unique contrairement à la méthode locale.

Cette étude a de plus permis de caractériser l'influence des paramètres des différente méthodes évaluées sur la solution.

Il ne reste plus maintenant qu'à conclure ces travaux et à tracer certaines perspectives.

Chapitre 7

Conclusion

Cette thèse s'inscrit dans le sillage des nombreuses recherches menées autour du problème d'estimation du flot optique. Elle se distingue par le fait d'avoir été menée dans un cadre pluridisciplinaire. Elle traite en effet d'un problème fondamental en traitement d'images (flot optique) auquel ont été appliqué des techniques de calcul scientifique (parallélisation) en visant une application expérimentale de la mécanique des fluide (PIV).

L'objectif fixé au début de ces travaux était de produire une méthode de calcul du flot optique précise, robuste et rapide. Le traitement d'images issues de la vélocimétrie par images de particules était l'application principalement visée.
Dans ce cadre particulier, nous avons évalué les deux grandes familles de méthodes existantes (par régularisation globale ou locale) afin d'observer leurs caractéristiques respectives.

Cette étude préliminaire nous a conduit à considérer le modèle CLG introduit par Bruhn *et al.* qui combine les régularisations locale et globale. L'examen de cette méthode nous a permis d'exhiber une approximation dans la mise en œuvre de warping du modèle original. Après une discussion et interprétation de cette approximation, nous avons proposé une formulation efficace permettant de la corriger que nous avons baptisé "*modified* CLG" (MCLG).
Cette formulation peut être considérée comme une justification de l'utilisation du warping dans CLG [Fezzani *et al.*, 2010].

À la suite de cette première étape, nous avons sélectionné un modèle employant la régularisation par variation totale. Nous étions alors face à un problème de minimisation d'énergie. Deux voies ont été explorées. Dans la première, nous avons considéré le problème sous sa forme continue. Ceci nous a donné accès à l'ensemble des outils de calcul variationnel. Nous avons ainsi été amené à résoudre des équations d'Euler-Lagrange.
Pour cela, en s'appuyant sur le fait que les données traitées dans ce problème (*i.e.* les images) sont disposées sur une grille cartésienne régulière, nous mettons en place une résolution à l'aide de schéma de différences finies plutôt que d'utiliser des éléments finis.

Pour la résolution des équations d'Euler-Lagrange, nous devions dans un premier temps discrétiser ces équations. Nous avons pour commencer étudié l'approche classique de discrétisation très peu décrite et discutée dans la littérature. Nous avons ainsi constaté qu'elle se basait sur la combinaison de schéma de

différences finies du premier ordre des opérateurs gradient et divergences. Le détail des opérations nécessaires pour l'obtension de cette discrétisation est fourni dans l'annexe A. Le lien avec la dérivation du critère discret y est également établi.
À la suite de quoi, nous avons proposé une méthode alternative pour la discrétisation de la partie régularisation en employant des schémas de différences finies d'opérateurs du second degrés. Cette approche c'est avérée plus efficace que le schéma de discrétisation classique pour restituer les discontinuités du champ de vecteurs comme nous l'avons prouvé dans le chapitre 6.
Nous avons ensuite développé une approche de résolution de type Picard-SOR par bloc 2×2 employant des linéarisations locales.

Pour accélérer ce solveur, une première voie basée sur l'élaboration d'une approche multigrille a été explorée. Dans un premier temps, nous avons développé un solveur multigrille linéaire [Wesseling, 1991] permettant de résoudre le problème employant une linéarisation globale du système. Cette approche c'est avérée peu adaptée pour la gestion du caractère fortement non linéaire des équations traitées. Par la suite, des tentatives d'élaboration d'un solveur multigrille non-linéaire (FAS [Trottenberg *et al.*, 2001] ou encore MNM [Yavneh et Dardyk, 2006]) ont été entreprises. Malheureusement, toutes ces tentatives ont échoué.

La seconde voie explorée pour la minimisation du critère considère le problème du point de vue discret et s'inspire de méthodes d'optimisation, basées sur le paradigme de "*variable splitting*", devenues populaires dans la communauté du traitement d'images.
Nous décrivons ce procédé dans le cadre général permettant ainsi la construction d'un tel solveur pour n'importe quel modèle d'estimation du flot optique par régularisation globale. Nous décrivons donc la méthode "split-Bregman" en la comparant à la méthode par pénalisation quadratique employée par Zach *et al.* ou encore Werlberger *et al.*

Les deux approches développées dans ces travaux ont été pensées de manière à profiter de la puissance des multiprocesseurs actuels, il a fallu pour cela paralléliser les algorithmes.
La résolution par variable splitting telle que présentée ne nécessite pas d'effort particulier pour sa parallélisation (grâce à l'utilisation de solveurs employant des opérations naturellement décorellées).
Par contre, pour la résolution par calcul variationnel, nous avons mis en place une méthode de "color labelling" pour permettre la décomposition du solveur en groupe d'opérations indépendantes les unes des autres ce qui permet leur traitement en paralléle.

Cette thèse a mené au développement de plusieurs codes parallèles sur deux architectures différentes : les processeurs multi-cœurs et les unités graphiques de calcul (GPU).

Lors de l'évaluation des algorithmes développés sur CPU, les expériences menées montrent que la méthode de variable splitting offre les meilleurs temps de calcul, mais la précision du résultat est moins bonne. Nous prouvons également que les solveurs par calcul variationnel bénéficient mieux de l'augmentation du nombre d'unités de calcul grâce à une meilleure exploitation de la mémoire cache des processeurs.
Les temps de calcul mesurés comparé à ceux affichés par les méthodes évaluées sur Middlebury montrent que nos algorithmes sont parmi les plus rapides : dans les conditions d'évaluation, l'algorithme $MCLG^{CV}_{ISO_{8ppv}}$ qui, parmis nos algoritmes, permet d'obtenir les meilleurs résultats, affiche un temps de calcul de 15.3 secondes pour des images de taille 460×640, ce qui le classe parmis les dix algorithmes les plus rapides.
Nous avons de plus montré que le multithreading sur un CPU récent doté de plusieurs cœurs à haute cadence pouvait rivaliser avec les performances d'un GPU pour la résolution par calcul variationnel.

Nous avons ensuite évalué notre approche sur différents types de données, prouvant ainsi la précision des résultats produits aussi bien sur des séquences classiques que sur des images de particules.
Ces expériences permettent de mettre en valeur l'approche MCLG par rapport à CLG. Ensuite, la méthode alternative de discrétisation des équations d'Euler-Lagrange développée dans cette thèse se distingue par sa capacité à mieux restituer les discontinuités du champ de vecteurs que l'approche classique. De plus, la méthode de résolution par variable splitting se révèle moins précise que celle par calcul variationnel.

Dans le contexte du traitement d'images PIV, nous procédons à une étude détaillée des caractéristiques de la combinaison des régularisations locale et globale. Nous prouvons notamment que MCLG permet d'obtenir, pour un niveau de bruit équivalent, une meilleure résolution spatiale que la méthode locale et cela pour différents cas de bruits. Lors de ces expériences, la solution de la méthode globale souffre d'un biais important que MCLG parvient à corriger.

Nous montrons de plus, à travers cette étude, l'influence des différents paramètres sur la qualité des résultats. Il en ressort notamment que la résolution spatiale de la méthode MCLG dépend essentiellement de la partie régularisation locale. En effet, dès qu'on utilise un écart type de gaussienne σ suppérieur à 1.33, le paramètre de régularisation α n'influence que très peu la résolution spatiale.

7.1 Perspectives

Dans la continuité de cette thèse, plusieurs travaux et voies de recherche peuvent être envisagés. Une première étape serait d'abord d'achever le portage sur GPU des différents algorithmes développés notamment la méthode de variable splitting qui, à cause de sa structure algorithmique se prête bien à la parallélisation

sur GPU et devrait mieux tirer profit de cette architecture que la méthode par calcul variationnel.
Nous avons d'autre part mis en évidence l'effet du goulot d'étranglement de Von Neumann (*c.f.* section 5.2.3.1) dans les versions CPU de nos codes qui est dû à la saturation de la mémoire cache. Il faudrait, pour atténuer cet effet, mettre en œuvre une résolution par bloc et ainsi accélérer les traitements sur l'architecture multi-cœurs.
Toujours dans le même souci d'accélération des calculs, nous pourrions envisager d'inclure nos algorithmes parallélisés à un grain fin dans une approche par décomposition de domaine. Ceci nous permettrait de profiter pleinement des machines disposant de plusieurs multiprocesseurs, ou encore d'envisager des architectures hétérogènes composées aussi bien de CPU que de GPU.
D'autre part, il est à noter que la convergence des différents solveurs n'est pas contrôlée dans l'état actuel des choses. En effet, le nombre d'itérations est fixé au lancement du programme. On pourrait envisager d'inclure un critère d'arrêt dans le processus itératif.

De plus, les algorithmes développés dans cette thèse ne prennent en compte que deux images succéssives. Une régularisation spatio-temporelle lorsqu'on dispose de plus de deux images dans une séquence pourrait être mise en œuvre.

D'autre part, un développement intéressant serait l'extension des méthodes développées à la stéréo-vision afin de permettre l'estimation de champs de déplacements en trois dimensions. Une telle évolution est possible à peu de frais car cette technique repose sur la reconstruction par triangulation d'un champ de vecteurs à trois composantes sur un plan à partir de deux estimations planes du flot optique [Leclaire *et al.*, 2011].
Par contre, l'adaptation de methodes d'estimation du flot optique à la tomographie respose elle sur la considération des équations sur un volume et non plus dans un plan. Le principe reste le même, mais on dispose cette fois-ci de trois inconnues, à savoir, les trois composantes du déplacement.

Pour finir, nous avons considéré dans ces travaux uniquement la régularisation par variation totale, mais nous pourrions envisager (à l'instar de Ruhnau *et al.* Ruhnau et Schnörr [2007]; Ruhnau *et al.* [2007]) de considérer une régularisation adaptée à la représentation de mouvement de fluides lors du traitement d'images de PIV. Dans le cadre tridimensionnel décrit au paragraphe précédents, l'élaboration d'un tel terme de régularisation serait plus naturel, car les équations qui régissent la mécaniques des fluides sont en général posées dans l'espace à trois dimensions.
La résolution par séparation de critère nous semble offrir dans ce contexte un cadre bien approprié. En effet, la séparation de la partie adéquation aux données de celle de régularisation permettrait un traitement plus facile de cette dernière.

Annexe A

Schéma classique pour la discrétisation de la partie régularisation

Sommaire

A.1 Détail du développement classique **95**
A.2 Dérivation du critère discret **96**
A.2.1 Différence avant 97
A.2.2 Différence arrière 97
A.2.3 Combinaison des différences avant et arrière 98
A.3 Conclusion **98**

Nous présentons dans cet annexe les détails de la discrétisation classique de la partie régularisation présentée dans la partie 3.2.1. Nous montrerons également la différence entre cette discrétisation et la dérivée du critère discret.

En adoptant les notations introduites à la partie 3.2.1, à savoir $(w, \tilde{w}) \in \{(u, v), (v, u)\}$ et
- $\Psi_{Ri,j} = \psi'(|\nabla w_{i,j}|^2)$ dans le cas anisotrope (3.7)-(3.8),
- $\Psi_{Ri,j} = \psi'(|\nabla w_{i,j}|^2 + |\nabla \tilde{w}_{i,j}|^2)$ dans le cas isotrope (3.4)-(3.5),

la partie régularisation des équations d'Euler-Lagrange, développées au chapitre 3, s'écrit en tout point de Ω

$$\mathrm{div}(\Psi_{Ri,j} \nabla w_{i,j}) = \partial_x(\Psi_{Ri,j} \partial_x w_{i,j}) + \partial_y(\Psi_{Ri,j} \partial_y w_{i,j}). \tag{A.1}$$

A.1 Détail du développement classique

Les dérivées partielles sont approchées par les différence finies avant ou arrière :

$$\partial_x f_{i,j} \approx D_x^+ f_{i,j} = f_{i,j+1} - f_{i,j} \tag{A.2}$$
$$\partial_x f_{i,j} \approx D_x^- f_{i,j} = f_{i,j} - f_{i,j-1} \tag{A.3}$$
$$\partial_y f_{i,j} \approx D_y^+ f_{i,j} = f_{i+1,j} - f_{i,j} \tag{A.4}$$
$$\partial_y f_{i,j} \approx D_y^- f_{i,j} = f_{i,j} - f_{i-1,j} \tag{A.5}$$

Le premier terme de l'addition dans (A.1) s'écrit

$$\partial_x(\Psi_{Ri,j}\partial_x w_{i,j}) = \frac{1}{2}\partial_x(\Psi_{Ri,j}\partial_x w_{i,j}) + \frac{1}{2}\partial_x(\Psi_{Ri,j}\partial_x w_{i,j}) \tag{A.6}$$

$$\approx \frac{1}{2}\big(D_x^+(\Psi_{Ri,j}D_x^- w_{i,j}) + D_x^-(\Psi_{Ri,j}D_x^+ w_{i,j})\big) \tag{A.7}$$

$$\approx \frac{1}{2}\big(\ \Psi_{Ri,j+1}(w_{i,j+1} - w_{i,j}) - \Psi_{Ri,j}(w_{i,j} - w_{i,j-1}) + \Psi_{Ri,j}(w_{i,j+1} - w_{i,j}) - \Psi_{Ri,j-1}(w_{i,j} - w_{i,j-1})\big) \tag{A.8}$$

$$\approx \frac{1}{2}\big(\ (\Psi_{Ri,j+1} + \Psi_{Ri,j})\cdot(w_{i,j+1} - w_{i,j}) - (\Psi_{Ri,j} + \Psi_{Ri,j-1})\cdot(w_{i,j} - w_{i,j-1})\big). \tag{A.9}$$

De même,

$$\partial_y(\Psi_{Ri,j}\partial_y w_{i,j}) \approx \frac{1}{2}\big(D_y^+(\Psi_{Ri,j}D_y^- w_{i,j}) + D_y^-(\Psi_{Ri,j}D_y^+ w_{i,j})\big) \tag{A.10}$$

$$\approx \frac{1}{2}\big(\ (\Psi_{Ri+1,j} + \Psi_{Ri,j})\cdot(w_{i+1,j} - w_{i,j}) - (\Psi_{Ri,j} + \Psi_{Ri-1,j})\cdot(w_{i,j} - w_{i-1,j})\big). \tag{A.11}$$

On retrouve ainsi la discrétisation classique du terme de régularisation (3.11)

$$\begin{aligned}\mathrm{div}(\Psi_{Ri,j}\nabla w_{i,j}) \approx \frac{1}{2}\big(\ &(\Psi_{Ri,j+1} + \Psi_{Ri,j})\cdot(w_{i,j+1} - w_{i,j}) - \\ &(\Psi_{Ri,j} + \Psi_{Ri,j-1})\cdot(w_{i,j} - w_{i,j-1}) + \\ &(\Psi_{Ri+1,j} + \Psi_{Ri,j})\cdot(w_{i+1,j} - w_{i,j}) - \\ &(\Psi_{Ri,j} + \Psi_{Ri-1,j})\cdot(w_{i,j} - w_{i-1,j})\big).\end{aligned} \tag{A.12}$$

Nous montrons par la suite la différence de cette discrétisation avec l'expression de la dérivée du critère considéré dans sa version discrète.

A.2 Dérivation du critère discret

Soit la version discrète de l'énergie (2.12) avec régularisation anisotrope et terme d'adéquation aux données quelconque :

$$E = \sum_{p,q}\psi\Big(\mathcal{G} * \big(I(\cdot + \mathbf{u}_{p,q}) - I(\cdot)\big)^2\Big) + \alpha\underbrace{\big(\psi(|Du_{p,q}|^2) + \psi(|Dv_{p,q}|^2)\big)}_{E_R} \tag{A.13}$$

où $D = (D_x, D_y)$ est le schéma de différences finies utilisé pour approcher le gradient ∇.
Minimiser E revient à annuler en tout point (i,j) la dérivée de E par rapport à $u_{i,j}$ et $v_{i,j}$:

$$\partial_{u_{i,j}}\psi\Big(\mathcal{G} * \big(I(\cdot + \mathbf{u}_{i,j}) - I(\cdot)\big)^2\Big) + \alpha\partial_{u_{i,j}}\sum_{p,q}\psi(|Du_{p,q}|^2) = 0 \tag{A.14}$$

$$\partial_{v_{i,j}}\psi\Big(\mathcal{G} * \big(I(\cdot + \mathbf{u}_{i,j}) - I(\cdot)\big)^2\Big) + \alpha\partial_{v_{i,j}}\sum_{p,q}\psi(|Dv_{p,q}|^2) = 0. \tag{A.15}$$

En employant le schéma de warping développé (2.9), les dérivées partielles de la partie adéquation aux données devient

$$\partial_{u_{i,j}}\psi(\mathcal{T}) = \psi'(\mathcal{T})\left(\mathcal{G} * \left[2D_xI(\cdot)\Big(D_xI^n(\cdot)u_{i,j}^{n+1} + D_yI^n(\cdot)v_{i,j}^{n+1} + \bar{D}_tI^n(\cdot)\Big)\right]\right) \tag{A.16}$$

$$\partial_{v_{i,j}}\psi(\mathcal{T}) = \psi'(\mathcal{T})\left(\mathcal{G} * \left[2D_yI(\cdot)\Big(D_xI^n(\cdot)u_{i,j}^{n+1} + D_yI^n(\cdot)v_{i,j}^{n+1} + \bar{D}_tI^n(\cdot)\Big)\right]\right) \tag{A.17}$$

où $\mathcal{T} = \mathcal{G} * \big(I(\cdot + \mathbf{u}_{i,j}) - I(\cdot)\big)^2$.

Nous allons étudier la dérivation de la partie régularisation E_R lorsque D est le schéma de différence finie avant ou arrière.

A.2.1 Différence avant

Supposons que D est défini à l'aide des différences finies avant :

$$D^+ = (D_x^+, D_y^+). \tag{A.18}$$

Considérons la dérivée de la partie régularisation E_R de l'énergie discrète (A.13) par rapport à $w_{i,j} \in \{u_{i,j}, v_{i,j}\}$.

$$\frac{\partial E_R}{\partial w_{i,j}} = \frac{\partial}{\partial w_{i,j}} \sum_{p,q} \psi(|D^+ w_{p,q}|^2) \tag{A.19}$$

$$= \frac{\partial}{\partial w_{i,j}} \Big(\psi(|D^+ w_{i,j}|^2) + \psi(|D^+ w_{i-1,j}|^2) + \psi(|D^+ w_{i,j-1}|^2)\Big) \tag{A.20}$$

$$\begin{aligned} = 2\Big(& -\Psi_{Ri,j}^+(D_x^+ w_{i,j} + D_y^+ w_{i,j}) + \\ & \Psi_{Ri-1,j}^+(D_y^+ w_{i-1,j}) + \\ & \Psi_{Ri,j-1}^+(D_x^+ w_{i,j-1})\Big) \end{aligned} \tag{A.21}$$

où

$$\Psi_{Ri,j}^+ = \psi'(|D^+ w_{i,j}|^2) \tag{A.22}$$

A.2.2 Différence arrière

En considérant cette fois-ci les différences finies arrières

$$D^- = (D_x^-, D_y^-), \tag{A.23}$$

on obtient de même

$$\begin{aligned} \frac{\partial E_R}{\partial w_{i,j}} = 2\Big(& \Psi_{Ri,j}^-(D_x^- w_{i,j} + D_y^- w_{i,j}) - \\ & \Psi_{Ri+1,j}^-(D_y^- w_{i+1,j}) - \\ & \Psi_{Ri,j+1}^-(D_x^- w_{i,j+1})\Big) \end{aligned} \tag{A.24}$$

où

$$\Psi_{Ri,j}^- = \psi'(|D^- w_{i,j}|^2) \tag{A.25}$$

A.2.3 Combinaison des différences avant et arrière

En considérant que E_R est une combinaison des expressions (A.21)-(A.24), et en remarquant que

$$D_x^+ w_{i,j-1} = D_x^- w_{i,j}, \tag{A.26}$$
$$D_y^+ w_{i-1,j} = D_y^- w_{i,j}, \tag{A.27}$$
$$D_x^- w_{i,j+1} = D_x^+ w_{i,j}, \tag{A.28}$$
$$D_y^- w_{i+1,j} = D_y^+ w_{i,j}, \tag{A.29}$$

on obtient

$$\frac{\partial E_R}{\partial w_{i,j}} = \frac{\partial}{\partial w_{i,j}} \frac{1}{2} \Big(\sum_{p,q} \psi'(|D^+ w_{p,q}|^2) + \sum_{p,q} \psi'(|D^- w_{p,q}|^2) \Big) \tag{A.30}$$

$$\begin{aligned} = \Big(& -D_x^+ w_{i,j} (\Psi^-_{Ri,j} + \Psi^-_{Ri,j+1}) \\ & -D_y^+ w_{i,j} (\Psi^-_{Ri,j} + \Psi^-_{Ri+1,j}) \\ & +D_x^- w_{i,j} (\Psi^+_{Ri,j} + \Psi^+_{Ri,j-1}) \\ & +D_y^- w_{i,j} (\Psi^+_{Ri,j} + \Psi^+_{Ri-1,j}) \Big) \end{aligned} \tag{A.31}$$

$$\begin{aligned} = -\Big(& (\Psi^-_{Ri,j} + \Psi^-_{Ri,j+1})(w_{i,j+1} - w_{i,j}) - \\ & (\Psi^+_{Ri,j} + \Psi^+_{Ri,j-1})(w_{i,j} - w_{i,j-1}) + \\ & (\Psi^-_{Ri,j} + \Psi^-_{Ri+1,j})(w_{i+1,j} - w_{i,j}) - \\ & (\Psi^+_{Ri,j} + \Psi^+_{Ri-1,j})(w_{i,j} - w_{i-1,j}) \Big). \end{aligned} \tag{A.32}$$

On obtient une formulation similaire pour le cas de la régularisation isotrope. Dans ce cas là $\Psi_R = \psi(|Du|^2 + |Dv|^2)$.

Remarque A.2.1 *Le facteur* $\frac{1}{2}$ *présent dans l'expression* (A.12) *compense le facteur 2 présent dans les dérivée du terme d'adéquation aux données de l'énergie discrète* (A.16) *et* (A.17). *De même, le signe "-" dans l'expresion* (A.32) *correspond au passage de l'expression* (A.1) *à gauche dans les équations d'Euler-Lagrange* (3.7)-(3.8) *et* (3.4)-(3.5).

A.3 Conclusion

Malgré les similitudes entre les expressions du schéma classique de discrétisation du terme de régularisation (A.12) et de la dérivée de la partie régularisation de l'énergie discrète (A.32), on a démontré ici que ces deux expressions sont en général différentes. En effet, la dérivation du critère discret produit un cas particulier de la discrétisation des équations d'Euler-Lagrange, car lorsque la version continue de l'énergie est considérée, le schéma de différences finies utilisé pour approcher les dérivées partielles de $w_{i,j}$ dans l'expression $\Psi_{R_{i,j}}$ n'est pas spécifié. Alors que si l'on considère la dérivation du critère discret, ces schémas de différences finies sont explicites dans les expressions $\Psi^+_{R_{i,j}}$ et $\Psi^-_{R_{i,j}}$.

Annexe B

Coefficients des système locaux induit par les méthodes variationnelles du chapitre 3

Sommaire

B.1 Régularisation anisotrope 100

B.1.1 Méthode classique 100

B.1.2 Méthode alternative 101

B.2 Régularisation isotrope 101

B.2.1 Méthode classique 101

B.2.2 Méthode alternative 102

La méthode de Picard développée au chapitre 3 donne localement lieu à différents systèmes linéaires sous la forme (3.26)-(3.27).

$$\begin{aligned}\mathcal{N}^u_{i,j}(\ & u^{n+1}_{i-1,j-1}, v^{n+1}_{i-1,j-1}, u^{n+1}_{i-1,j}, v^{n+1}_{i-1,j}, u^{n+1}_{i-1,j+1}, v^{n+1}_{i-1,j+1}, \\ & u^{n+1}_{i,j-1}, v^{n+1}_{i,j-1}, u^{n+1}_{i,j}, v^{n+1}_{i,j}, u^{n+1}_{i,j+1}, v^{n+1}_{i,j+1}, \\ & u^{n+1}_{i+1,j-1}, v^{n+1}_{i+1,j-1}, u^{n+1}_{i+1,j}, v^{n+1}_{i+1,j}, u^{n+1}_{i+1,j+1}, v^{n+1}_{i+1,j+1}\ ) = 0,\end{aligned} \tag{B.1}$$

$$\begin{aligned}\mathcal{N}^v_{i,j}(\ & u^{n+1}_{i-1,j-1}, v^{n+1}_{i-1,j-1}, u^{n+1}_{i-1,j}, v^{n+1}_{i-1,j}, u^{n+1}_{i-1,j+1}, v^{n+1}_{i-1,j+1}, \\ & u^{n+1}_{i,j-1}, v^{n+1}_{i,j-1}, u^{n+1}_{i,j}, v^{n+1}_{i,j}, u^{n+1}_{i,j+1}, v^{n+1}_{i,j+1}, \\ & u^{n+1}_{i+1,j-1}, v^{n+1}_{i+1,j-1}, u^{n+1}_{i+1,j}, v^{n+1}_{i+1,j}, u^{n+1}_{i+1,j+1}, v^{n+1}_{i+1,j+1}\ ) = 0.\end{aligned} \tag{B.2}$$

Cette annexe liste les systèmes des différentes méthodes développées. On utilise pour cela la notation suivante :

$$\begin{Bmatrix} c_{-1,-1} & c_{-1,0} & c_{-1,1} \\ c_{0,-1} & c_{0,0} & c_{0,1} \\ c_{1,-1} & c_{1,0} & c_{1,1} \end{Bmatrix} w = \sum_{k=-1}^{1} \sum_{l=-1}^{1} c_{k,l} w_{i+k,j+l}. \tag{B.3}$$

Un ensemble de définitions est introduit également :

$$\Psi_D = \psi'\Big(\mathcal{G} * (\mathbf{u}^{n+1,k}(\mathbf{x})\bar{\nabla}^T I^n(\cdot)\bar{\nabla} I^n(\cdot)\mathbf{u}^{n+1,k}(\mathbf{x})^T)\Big), \tag{B.4}$$

$$J_{11} = \Psi_D \cdot \Big(\mathcal{G} * (\partial_x I^n)^2\Big), \tag{B.5}$$

$$J_{22} = \Psi_D \cdot \Big(\mathcal{G} * (\partial_y I^n)^2\Big), \tag{B.6}$$

$$J_{12} = \Psi_D \cdot \Big(\mathcal{G} * (\partial_x I^n \cdot \partial_y I^n)\Big), \tag{B.7}$$

$$J_{13} = \Psi_D \cdot \Big(\mathcal{G} * (\partial_x I^n \cdot \bar{\partial}_t I^n)\Big), \tag{B.8}$$

$$J_{23} = \Psi_D \cdot \Big(\mathcal{G} * (\partial_y I^n \cdot \bar{\partial}_t I^n)\Big). \tag{B.9}$$

B.1 Régularisation anisotrope

On traite de la résolution des équations d'Euler-Lagrange (3.7)-(3.8) avec terme de régularisation anisotrope.

B.1.1 Méthode classique

$$\begin{Bmatrix} 0 & \mathcal{R}^u_{-1,0} & 0 \\ \mathcal{R}^u_{0,-1} & J11 + \mathcal{SR}^u & \mathcal{R}^u_{0,1} \\ 0 & \mathcal{R}^u_{1,0} & 0 \end{Bmatrix} u^{n+1,k+1} + \begin{Bmatrix} 0 & 0 & 0 \\ 0 & J12 & 0 \\ 0 & 0 & 0 \end{Bmatrix} v^{n+1,k+1} + J13 = 0 \tag{B.10}$$

$$\begin{Bmatrix} 0 & \mathcal{R}^v_{-1,0} & 0 \\ \mathcal{R}^v_{0,-1} & J22 + \mathcal{SR}^v & \mathcal{R}^v_{0,1} \\ 0 & \mathcal{R}^v_{1,0} & 0 \end{Bmatrix} v^{n+1,k+1} + \begin{Bmatrix} 0 & 0 & 0 \\ 0 & J12 & 0 \\ 0 & 0 & 0 \end{Bmatrix} u^{n+1,k+1} + J23 = 0 \tag{B.11}$$

où

$$\Psi^*_R = \alpha\psi'(|\nabla *^{n+1,k}|^2) \qquad * \in \{u, v\}, \tag{B.12}$$

$$\mathcal{R}^*_{k,l} = \Psi^*_{Ri,j} + \Psi^*_{Ri+k,j+l} \qquad * \in \{u, v\}, \tag{B.13}$$

$$\mathcal{SR}^* = \mathcal{R}^*_{-1,0} + \mathcal{R}^*_{0,-1} + \mathcal{R}^*_{0,1} + \mathcal{R}^*_{1,0} \qquad * \in \{u, v\}. \tag{B.14}$$

B.1.2 Méthode alternative

On notera

$$\bar{\Psi}_R^* = \alpha \frac{\psi'(|\nabla *^{n+1,k}|^2)}{|\nabla *^{n+1,k}|^2 + \varepsilon} \qquad * \in \{u, v\}, \tag{B.15}$$

$$\mathcal{R}_{xx}^* = ((\partial_y *^{n+1,k})^2 + \varepsilon) \cdot \bar{\Psi}_R^* \qquad * \in \{u, v\}, \tag{B.16}$$

$$\mathcal{R}_{yy}^* = ((\partial_x *^{n+1,k})^2 + \varepsilon) \cdot \bar{\Psi}_R^* \qquad * \in \{u, v\}, \tag{B.17}$$

$$\mathcal{R}_{xy}^* = (2(\partial_x *^{n+1,k} \cdot \partial_y *^{n+1,k}) + \varepsilon) \cdot \bar{\Psi}_R^* \qquad * \in \{u, v\}, \tag{B.18}$$

$$\bar{\mathcal{S}\mathcal{R}}^* = 2(\mathcal{R}_{xx}^* + \mathcal{R}_{yy}^*) \qquad * \in \{u, v\} \tag{B.19}$$

pour obtenir le système suivant :

$$\begin{Bmatrix} \mathcal{R}_{xy}^u & \mathcal{R}_{yy}^u & \mathcal{R}_{xy}^u \\ \mathcal{R}_{xx}^u & J11 + \bar{\mathcal{S}\mathcal{R}}^u & \mathcal{R}_{xx}^u \\ \mathcal{R}_{xy}^u & \mathcal{R}_{yy}^u & \mathcal{R}_{xy}^u \end{Bmatrix} u^{n+1,k+1} + \begin{Bmatrix} 0 & 0 & 0 \\ 0 & J12 & 0 \\ 0 & 0 & 0 \end{Bmatrix} v^{n+1,k+1} + J13 = 0 \tag{B.20}$$

$$\begin{Bmatrix} \mathcal{R}_{xy}^v & \mathcal{R}_{yy}^v & \mathcal{R}_{xy}^v \\ \mathcal{R}_{xx}^v & J22 + \bar{\mathcal{S}\mathcal{R}}^v & \mathcal{R}_{xx}^v \\ \mathcal{R}_{xy}^v & \mathcal{R}_{yy}^v & \mathcal{R}_{xy}^v \end{Bmatrix} v^{n+1,k+1} + \begin{Bmatrix} 0 & 0 & 0 \\ 0 & J12 & 0 \\ 0 & 0 & 0 \end{Bmatrix} u^{n+1,k+1} + J23 = 0. \tag{B.21}$$

B.2 Régularisation isotrope

On traite ici le cas des équations d'Euler-Lagrange (3.4)-(3.5) avec terme de régularisation isotrope.

B.2.1 Méthode classique

$$\begin{Bmatrix} 0 & \mathcal{R}_{-1,0} & 0 \\ \mathcal{R}_{0,-1} & J11 + \mathcal{S}\mathcal{R} & \mathcal{R}_{0,1} \\ 0 & \mathcal{R}_{1,0} & 0 \end{Bmatrix} u^{n+1,k+1} + \begin{Bmatrix} 0 & 0 & 0 \\ 0 & J12 & 0 \\ 0 & 0 & 0 \end{Bmatrix} v^{n+1,k+1} + J13 = 0 \tag{B.22}$$

$$\begin{Bmatrix} 0 & \mathcal{R}_{-1,0} & 0 \\ \mathcal{R}_{0,-1} & J22 + \mathcal{S}\mathcal{R} & \mathcal{R}_{0,1} \\ 0 & \mathcal{R}_{1,0} & 0 \end{Bmatrix} v^{n+1,k+1} + \begin{Bmatrix} 0 & 0 & 0 \\ 0 & J12 & 0 \\ 0 & 0 & 0 \end{Bmatrix} u^{n+1,k+1} + J23 = 0 \tag{B.23}$$

où

$$\Psi_R = \psi'(|\nabla u^{n+1,k}|^2 + |\nabla v^{n+1,k}|^2), \tag{B.24}$$

$$\mathcal{R}_{k,l} = \Psi_{Ri,j} + \Psi_{Ri+k,j+l}, \tag{B.25}$$

$$\mathcal{S}\mathcal{R} = \mathcal{R}_{-1,0} + \mathcal{R}_{0,-1} + \mathcal{R}_{0,1} + \mathcal{R}_{1,0}. \tag{B.26}$$

B.2.2 Méthode alternative

En notant

$$\bar{\Psi}_R = \alpha \frac{\psi'(|\nabla u^{n+1,k}|^2 + |\nabla v^{n+1,k}|^2)}{|\nabla u^{n+1,k}|^2 + |\nabla v^{n+1,k}|^2 + \varepsilon}, \tag{B.27}$$

$$\mathcal{R}^{\star\star}_{xx} = ((|\nabla \star^{n+1,k}|^2 + \partial_y *^{n+1,k})^2 + \varepsilon) \cdot \bar{\Psi}_R \qquad *\star \in \{uv, vu\}, \tag{B.28}$$

$$\mathcal{R}^{\star\star}_{yy} = ((|\nabla \star^{n+1,k}|^2 + \partial_x *^{n+1,k})^2 + \varepsilon) \cdot \bar{\Psi}_R \qquad *\star \in \{uv, vu\}, \tag{B.29}$$

$$\bar{\mathcal{R}}^{*}_{xy} = (2(\partial_x *^{n+1,k} \cdot \partial_y *^{n+1,k}) + \varepsilon) \cdot \bar{\Psi}_R \qquad * \in \{u, v\}, \tag{B.30}$$

$$\mathcal{SR}^{\star\star} = 2(\mathcal{R}^{\star\star}_{xx} + \mathcal{R}^{\star\star}_{yy}) \qquad *\star \in \{uv, vu\}, \tag{B.31}$$

$$\mathcal{R}_{xx} = ((\partial_x u \cdot \partial_x v + \varepsilon) \cdot \bar{\Psi}_R, \tag{B.32}$$

$$\mathcal{R}_{yy} = ((\partial_y u \cdot \partial_y v + \varepsilon) \cdot \bar{\Psi}_R, \tag{B.33}$$

$$\begin{aligned} \mathcal{R}_{xy} = (\ & \partial_x u^{n+1,k} \cdot \partial_y v^{n+1,k} + \\ & \partial_y u^{n+1,k} \cdot \partial_x v^{n+1,k} + \varepsilon) \cdot \bar{\Psi}_R, \end{aligned} \tag{B.34}$$

$$\bar{\mathcal{SR}} = 2(\mathcal{R}_{xx} + \mathcal{R}_{yy}), \tag{B.35}$$

l'équation considérée s'écrit

$$\begin{Bmatrix} \bar{\mathcal{R}}^{u}_{xy} & \mathcal{R}^{uv}_{yy} & \bar{\mathcal{R}}^{u}_{xy} \\ \mathcal{R}^{uv}_{xx} & J11 + \mathcal{SR}^{uv} & \mathcal{R}^{uv}_{xx} \\ \bar{\mathcal{R}}^{u}_{xy} & \mathcal{R}^{uv}_{yy} & \bar{\mathcal{R}}^{u}_{xy} \end{Bmatrix} u^{n+1,k+1} + \begin{Bmatrix} \mathcal{R}_{xy} & \mathcal{R}_{yy} & \mathcal{R}_{xy} \\ \mathcal{R}_{xx} & J12 + \bar{\mathcal{SR}} & \mathcal{R}_{xx} \\ \mathcal{R}_{xy} & \mathcal{R}_{yy} & \mathcal{R}_{xy} \end{Bmatrix} v^{n+1,k+1} + J13 = 0 \tag{B.36}$$

$$\begin{Bmatrix} \bar{\mathcal{R}}^{v}_{xy} & \mathcal{R}^{vu}_{yy} & \bar{\mathcal{R}}^{v}_{xy} \\ \mathcal{R}^{vu}_{xx} & J22 + \mathcal{SR}^{vu} & \mathcal{R}^{vu}_{xx} \\ \bar{\mathcal{R}}^{v}_{xy} & \mathcal{R}^{vu}_{yy} & \bar{\mathcal{R}}^{v}_{xy} \end{Bmatrix} v^{n+1,k+1} + \begin{Bmatrix} \mathcal{R}_{xy} & \mathcal{R}_{yy} & \mathcal{R}_{xy} \\ \mathcal{R}_{xx} & J12 + \bar{\mathcal{SR}} & \mathcal{R}_{xx} \\ \mathcal{R}_{xy} & \mathcal{R}_{yy} & \mathcal{R}_{xy} \end{Bmatrix} u^{n+1,k+1} + J23 = 0. \tag{B.37}$$

Annexe C

Décomposition de domaine

Sommaire

C.1 Approche naïve **103**
C.2 Méthode de Schwarz avec recouvrement **104**

C.1 Approche naïve

Les différents algorithmes développés aux chapitres 3 et 4 peuvent être inclus dans une méthode par décomposition de domaine simple.
En effet, les méthodes précitées ne font intervenir au plus que les 8 plus proches voisins du points considéré dans le domaine image Ω. En découpant l'image en plusieurs parties, et en échangeant les éléments nécessaires dans les calculs à l'interface, les différents sous domaines peuvent être traité en parallèle.

La Figure C.2 montre une configuration intuitive pour la décomposition de Ω en sous domaines. On y voit l'interface entre deux parties Ω_1 et Ω_2. Les connections se font entre les lignes $l-1$ et l. Pour pouvoir traiter simultanément Ω_1 et Ω_2, on procédera comme suit :

1. les processeurs 1 et 2 traitent simultanément et respectivement Ω_1 et Ω_2,
2. synchronisation des processeurs 1 et 2,
3. processeur 1 envoie ligne $l-1$ à processeur 2,
4. processeur 2 envoie ligne l à processeur 1.

Cependant, pour garantir l'équivalence entre la solution sur un seul domaine et celle sur plusieurs domaines on devra prendre soin d'échanger les données à chaque étapes où c'est nécessaire. Par exemple, pour la méthode par calcul variationnel avec coloration de maille, les communications se feront à chaque fois que l'on traite les points d'une même couleur.

Ce genre de mise en œuvre entraîne un taux de communication important. Si on dispose par exemple de 12 threads, on pourrait soit appliquer une approche SIMD pure dans laquelle les 12 threads traiteraient l'ensemble des données, soit appliquer une approche MIMD avec 4 sous-domaines (par exemple) dans laquelle 4 groupes

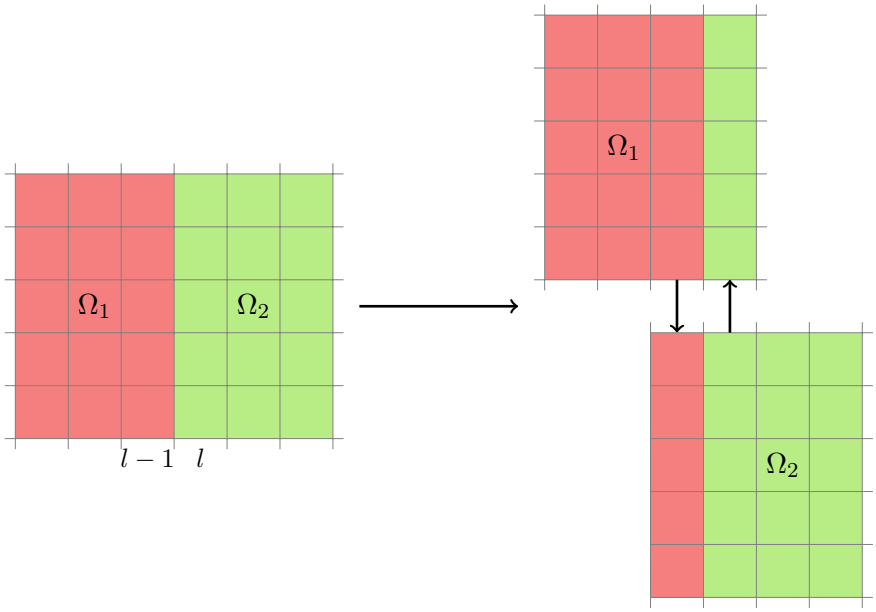

FIGURE C.1 – Interface entre deux sous-domaines Ω_1 et Ω_2 et illustration des communications nécessaires.

de 3 threads traiteraient simultanément chacun un quart des données.
Pour que l'approche à mémoire partagée soit rentable par rapport à celle SIMD pure, il faut que le temps de communication entre les sous-domaines soit inférieur à la différence des temps à volume de données constants par threads (FIGURE 5.6) pour 12 et 3 threads.

C.2 Méthode de Schwarz avec recouvrement

La méthode de schwartz [Lions, 1988] additive qui stipule d'effectuer les communications décrites précédemment non pas à chaque itération de solveur mais à intervalle plus grand.
Cette approche, contrairement à l'approche naïve, ne permet pas de retrouver les même solutions à chaque itération. Elle converge de ce fait moins rapidement.
Cette convergence peut être améliorée en employant une zone de recouvrement supérieur à une ligne. Ceci minimise l'influence des effets de bord. Cependant, on effectue dans ce cas là des calculs redondants.

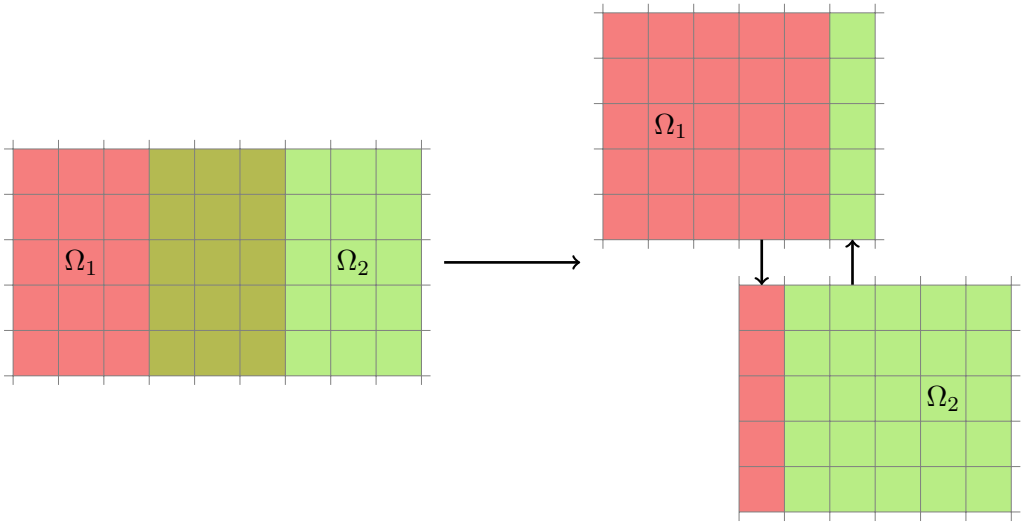

FIGURE C.2 – Illustration de la méthode de Schwarz avec recouvrement.

Liste des symboles

$4ppv$ Quatre plus proches voisins, page 52

$8ppv$ Huit plus proches voisins, page 52

α Paramètre de régularisation, page 7

$\mathcal{G}(x, y)$ Fonction gaussienne centrée en (x, y), page 6

∇ Gradient spatial (∂_x, ∂_y), page 4

∇_3 $(\partial_x, \partial_y, \partial_t)$, page 6

Ω Domaine image, page 2

$\partial^2_{*\star}$ $\frac{\partial^2}{\partial * \partial \star}$, page 4

∂_* $\frac{\partial}{\partial *}$, page 4

σ Écart-type de foncion gaussienne, page 55

$\mathbf{u}(\mathbf{x})$ Flot optique $(u(\mathbf{x}), v(\mathbf{x}), 1)$, page 5

$\mathcal{V}(x, y)$ Voisinage du point $(x, y) \in \Omega$, page 2

$\mathbf{x}$ Coordonnées spatio-temporelles $(x, y, t) \in \Omega \times \mathbb{R}$, page 5

$ANISO$ Anisotrope, page 52

CV Calcul variationnel, page 52

I Intensité lumineuse, page 2

I^n $I(\mathbf{x} + \mathbf{u}^n(\mathbf{x}))$: image interpolée à l'aide du champ de vecteurs $\mathbf{u}^n$, page 26

ISO Isotrope, page 52

R Rayon de la fenêtre utilisé dans la méthode locale, page 70

VS *Variable splitting, page 52*

CLG *Combined Local Global method*, page 19

CPU *Central processing unit, page 52*

CUDA *Compute Unified Device Architecture, page 52*

div Divergence $(\partial_x \cdot + \partial_y \cdot)$, page 35

GPU Carte graphique (*Graphics Processing Unit*), page 11

MCLG *Modified Combined Local Global method, page 28*

MIMD *Multiple instruction multiple data, page 11*

PIV *Particle Image Velocimetry, page 2*

SIMD *Single instruction multiple data, page 11*

TV Variation totale (*Total Variation*), page 8

UC Unité de calcul, page 10

Bibliographie

L. Adams et J. Ortega : A multi-color SOR method for parallel computation. *In Proc. 1982 International Conference on Parallel Processing, Bellaire, MI*, p. 53–58. Citeseer, 1982. 40

M. Afonso, J. Bioucas-Dias et M. Figueiredo : Fast image recovery using variable splitting and constrained optimization. *Image Processing, IEEE Transactions on*, 19(9):2345–2356, 2010. 43, 44, 46, 48

L. Alvarez, J. Esclarin, M. Lefebure et J. Sanchez : A pde model for computing the optical flow. *In XVI congreso de ecuaciones diferenciales y aplicaciones*, p. 1349–1356, 1999. 8, 34

P. Anandan : A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, jan 1989. 9

G. Aubert, R. Deriche et P. Kornprobst : Computing optical flow via variational techniques. *SIAM Journal on Applied Mathematics*, 60(1):156–182, 1999. 8, 44

G. Aubert et P. Kornprobst : A Mathematical Study of the Relaxed Optical Flow Problem in the Space $BV(\Omega)$. *SIAM Journal on Mathematical Analysis*, 30:1282, 1999. 8

G. Aubert et P. Kornprobst : *Mathematical problems in image processing.* Springer Science+ Business Media, LLC, 2006. 5

J. Backus : Can programming be liberated from the von Neumann style ? : a functional style and its algebra of programs. *Communication of the ACM*, 21 (8):613–641, 1978. 59

S. Baker, D. Scharstein, J. Lewis, S. Roth, M. Black et R. Szeliski : A database and evaluation methodology for optical flow. Rap. tech. MSR-TR-2009-179, Microsoft Research, 2009. 3, 17, 69

R. Bayt et K. Breuer : Fabrication and testing of micron-sized cold-gas thrusters. *In Micropropulsion for Small Spacecraft*, vol. 1, p. 381, 2000. ISBN 1563474484. 12

D. P. Bertsekas : *Constrained Optimization and Lagrange Multiplier Methods.* Athena Scientific, 1996. 44, 46

D. Bertsekas, W. Hager et O. Mangasarian : *Nonlinear programming.* Athena Scientific Belmont, MA, 1999. 44, 45

M. J. Black et P. Anandan : Robust dynamic motion estimation over time. *In IEEE computer society conference on computer vision and patern recognition*, p. 296–302, 1991. 9

M. Black : *Robust incremental optical flow.* Thèse de doctorat, Yale University, New Haven, CT, 1992. 8

M. Black et P. Anandan : The robust estimation of multiple motions : Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding*, 63 (1):75–104, 1996. 9, 34

L. Blanc-Feraud, M. Barlaud et T. Gaidon : Motion estimation involving discontinuities in multiresolution scheme. *Optical Engineering*, 32:1475, 1993. 8, 34

F. Bornemann et P. Deuflhard : The cascadic multigrid method for elliptic problems. *Numerische Mathematik*, 75(2):135–152, 1996. 52

L. Bregman : The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming* 1. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967. 44, 46

C. Brossard, J. Monnier, P. Barricau, F. Vandernoot, Y. Le Sant, F. Champagnat et G. Le Besnerais : Principles and Applications of Particle Image Velocimetry. *AerospaceLab*, 1(1), 2009. 12

A. Bruhn, J. Weickert, C. Feddern, T. Kohlberger et C. Schnorr : Variational optical flow computation in real time. *IEEE Transactions on Image Processing*, 14(5):608–615, 2005a. 38

A. Bruhn, J. Weickert, T. Kohlberger et C. Schnörr : A multigrid platform for real-time motion computation with discontinuity-preserving variational methods. *Int. J. Comput. Vision*, 70(3):257–277, 2006. ISSN 0920-5691. 35, 38

A. Bruhn, J. Weickert et C. Schnörr : Lucas/kanade meets horn/schunck : Combining local and global optic flow methods. *International Journal of Computer Vision*, 61:211–231, 2005b. i, iii, 16, 19, 25, 29, 34

H. Cartan : *Cours de calcul différentiel.* Hermann, 1977. 7

A. Chambolle et T. Pock : A first-order primal-dual algorithm for convex problems with applications to imaging. 2010. 47

A. Chambolle : An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20(1-2):89–97, 2004. ISSN 0924-9907. 12, 48

F. Champagnat, A. Plyer, G. Le Besnerais, B. Leclaire, S. Davoust et Y. Le Sant : Accelerating PIV computation speed using highly parallel iterative correlation maximization algorithm. *To appear in Experiments in fluids*, 2011. 7, 16

F. Champagnat, A. Plyer, G. Le Besnerais, B. Leclaire et Y. Le Sant : How to calculate dense PIV vector fields at video rates. *In Proceedings of 8th International Symposium on Particle Image Velocimetry - PIV09*, Melbourne, 2009. 11, 15, 20

T. Chan et P. Mulet : On the convergence of the lagged diffusivity fixed point method in total variation image restoration. *SIAM Journal on Numerical Analysis*, 36(2):354–367, 1999. 38

B. Chapman, G. Jost et R. Van Der Pas : *Using OpenMP : portable shared memory parallel programming.* The MIT Press, 2008. ISBN 0262533022. 52, 56

I. Cohen : Nonlinear variational method for optical flow computation. *In Proceeding of the 8th SCIA*, p. 523–530, jun 1993. 8, 34

T. Corpetti, E. Mémin et P. Pérez : Dense estimation of fluid flows. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(3):365–380, 2002. 15

T. Corpetti, D. Heitz, G. Arroyo, E. Mémin et A. S. Cruz : Fluid experimental flow estimation based on optical-flow scheme. *Springer Berlin/Heidelberg, Experiments in Fluids*, 40(1):80–97, jan 2006. 15

J. Dongarra *et al.* : Basic linear algebra subprograms technical forum standard. *International Journal of High Performance Applications and Supercomputing*, 16 (1):1–111, 2002. 56

J. Eckstein et D. Bertsekas : On the Douglas—Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, 1992. 47

G. Elsinga, F. Scarano, B. Wieneke et B. Van Oudheusden : Tomographic particle image velocimetry. *Experiments in Fluids*, 41(6):933–947, 2006. ISSN 0723-4864. 12

E. Esser : Applications of Lagrangian-based alternating direction methods and connections to split Bregman. *CAM report*, 9:31, 2009. 47

R. Fezzani, F. Champagnat et G. Le Besnerais : Clarifying the implementation of warping in the combined local global method for optic flow computation. *In 18th European Signal Processing Conference (EUSIPCO)*, p. 1321–1325. EURASIP, 2010. 28, 91

D. Geman et G. Reynolds : Constrained restoration and the recovery of discontinuities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14 (3):367–383, 1992. 44

D. Geman et C. Yang : Nonlinear image recovery with half-quadratic regularization. *Image Processing, IEEE Transactions on*, 4(7):932–946, 1995. 44

T. Goldstein et S. Osher : The split Bregman method for L1 regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009. 43, 44, 49

G. Golub et C. Van Loan : *Matrix computations.* Johns Hopkins Univ Pr, 1996. 38

S. Grewenig, J. Weickert et A. Bruhn : From box filtering to fast explicit diffusion. *Pattern Recognition*, p. 533–542, 2010. 12

P. Gwosdek, H. Zimmer, S. Grewenig, A. Bruhn et J. Weickert : A highly efficient GPU implementation for variational optic flow based on the Euler-Lagrange framework. *In Proc. 2010 ECCV Workshop on Computer Vision with GPUs*, 2010. 12

D. Heitz, P. Héas, E. Mémin et J. Carlier : Dynamic consistent correlation-variational approach for robust optical flow estimation. *Experiments in fluids*, 45 (4):595–608, 2008. ISSN 0723-4864. 15

D. Heitz, E. Mémin et C. Schnörr : Variational fluid flow measuremts from image sequences : synopsis and perspectives. *Experiments in fluids*, 48(3):369–393, 2010. 7, 15

B. K. P. Horn et B. G. Schunck : Determining optical flow. *Artificial intelligence*, 17:185–203, 1981. 3, 4, 7, 8

Y. Huang, M. Ng et Y. Wen : A fast total variation minimization method for image restoration. *Multiscale Modeling & Simulation*, 7:774, 2008. 43

P. Huber : Robust regression : asymptotics, conjectures and Monte Carlo. *The Annals of Statistics*, 1(5):799–821, 1973. ISSN 0090-5364. 48

J. Idier : Convex half-quadratic criteria and interacting auxiliary variables for image restoration. *Image Processing, IEEE Transactions on*, 10(7):1001–1009, 2001. 44

T. Kohlberger, C. Schnorr, A. Bruhn et J. Weickert : Domain decomposition for variational optical-flow computation. *Image Processing, IEEE Transactions on*, 14(8):1125–1137, 2005. ISSN 1057-7149. 11, 12

A. Kumar, A. Tannenbaum et G. Balas : Optical flow : a curve evolution approach. *IEEE Transactions on Image Processing*, 5:598–610, 1996. 8

G. Le Besnerais et F. Champagnat : Dense optical flow by iterative local window registration. *In IEEE International Conference on Image Processing*, 2005. 9, 26

G. Le Besnerais, F. Champagnat, A. Plyer, R. Fezzani, B. Leclaire et Y. L. Sant : Advenced processing methods for image-based displacement measurement. *Aerospace Lab*, 1(1), 2009. 38

B. Leclaire, Y. Le Sant, S. Davoust, G. Le Besnerais et F. Champagnat : Folki-spiv : a new, ultra-fast approach for stereo piv. *To appear in Experiments in fluids*, 2011. 94

B. Lecordier et J. Westerweel : The EUROPIV synthetic image generator (SIG). *In Particle image velocimetry : recent improvements. Proceedings of the EUROPIV*, vol. 2, 2004. 20

P. Lions : On the Schwarz alternating method I. *Domain decomposition methods for partial differential equations*, p. 1–41, 1988. 104

B. D. Lucas et T. Kanade : An iterative image registration technique with an application to stereo vision. *In Seventh international joint conference on artificial intelligence*, p. 674–679, 1981. 3, 5

E. Mémin et P. Pérez : Optical flow estimation and object-based segmentation with robust techniques. *IEEE Trans. on Image Processing*, 7(5):703–719, May 1998. 44

H. H. Nagel et W. Enkelmann : An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(5):565–593, 1986. ISSN 0162-8828. 8, 34

J. Nocedal et S. Wright : *Numerical optimization*. Springer verlag, 1999. 44, 45

C. Nvidia : Compute Unified Device Architecture Programming Guide. *NVIDIA : Santa Clara, CA*, 2007. 52, 63, 65

C. Nvidia : Tuning CUDA Applications for FERMI. *NVIDIA : Santa Clara, CA*, 2010. 65

N. Papenberg, A. Bruhn, T. Brox, S. Didas et J. Weickert : Highly accurate optic flow computation with theoretically justified warping. *International Journal of Computer Vision*, 67(2):141–158, 2006. 8, 9, 10, 26, 27, 34

Y. Penel : Approche parallèle pour l'approximation du flot optique par méthodes variatrionnelles. Rap. tech., ONERA, the French Aerospace Lab, 2007. 11, 12

M. Pilgrim : *Dive into Python*. Apress, 2004. URL `http://www.diveintopython.org/`. 52

G. Quenot, J. Pakleza et T. Kowalewski : Particle image velocimetry with optical flow. *Experiments in fluids*, 25(3):177–189, 1998. 15, 16

M. Raffel, C. Willert et S. Wereley : *Particle image velocimetry : a practical guide*. Springer Verlag, 2007. 2, 12, 14

L. I. Rudin, S. Osher et E. Fatemi : Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992. ISSN 0167-2789. 8, 47, 48

P. Ruhnau, T. Kohlberger, C. Schnörr et H. Nobach : Variational optical flow estimation for particle image velocimetry. *Experiments in Fluids*, 38(1):21–32, 2005. 15, 16, 20

P. Ruhnau et C. Schnörr : Optical stokes flow estimation : an imaging-based control approach. *Experiments in Fluids*, 42(1):61–78, 2007. 15, 94

P. Ruhnau, A. Stahl et C. Schnörr : Variational estimation of experimental fluid flows with physics-based spatio-temporal regularization. *Measurement Science and Technology*, 18:755, 2007. 15, 94

Y. Saad : *Iterative methods for sparse linear systems.* Society for Industrial Mathematics, 2003. 38, 40

F. Scarano et M. Riethmuller : Advances in iterative multigrid PIV image processing. *Experiments in Fluids*, 29:51–60, 2000. ISSN 0723-4864. 83

S. Setzer : Split Bregman algorithm, Douglas-Rachford splitting and frame shrinkage. *Scale space and variational methods in computer vision*, p. 464–476, 2009. 47

M. Stanislas, K. Okamoto et C. Kahler : Main results of the first international PIV challenge. *Measurement Science and Technology*, 14(10):63–89, 2003. 20

M. Stanislas, K. Okamoto, C. J. Kähler et J. Westerweel : Main results of the second international piv challenge. *Experiments in Fluids*, 39:170–191, 2005. xi, 14, 86

F. Steinbruecker, T. Pock et D. Cremers : Advanced data terms for variational optic flow estimation. *In Vision, Modeling, and Visualization Workshop*, Braunschweig, Germany, 2009. 47

D. Suter : Motion estimation and vector splines. *In Computer Vision and Pattern Recognition, 1994 IEEE Computer Society Conference*, p. 939–942. IEEE, 2002. 15

X. Tai et C. Wu : Augmented Lagrangian method, dual methods and split Bregman iteration for ROF model. *Scale Space and Variational Methods in Computer Vision*, p. 502–513, 2009. 47

A. Tikhonov : On the stability of inverse problems. *In CR (Dokl.) Acad. Sci. URSS, n. Ser*, vol. 39, p. 176–179, 1943. 7

M. Tistarelli : Computation of coherent optical flow by using multiple constraints. *In Computer Vision, 1995. Proceedings., Fifth International Conference on*, p. 263–268. IEEE, 2002. 4

O. Tretiak et L. Pastor : Velocity estimation from image sequences with second order differential operators. *In International Conference on pattern recognition*, p. 16–19, 1984. 4

U. Trottenberg, C. W. Oosterlee et A. Schüller : *Multigrid*. Academic Pr, 2001. 38, 40, 92

R. Tsuchiyama, T. Nakamura, T. Iizuka, A. Asahara et S. Miki : The OpenCL Programming Book. *Fixstars Corporation*, 2010. 63

C. Vogel et M. Oman : Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, 1996. 38

Y. Wang, J. Yang, W. Yin et Y. Zhang : A new alternating minimization algorithm for total variation image reconstruction. *SIAM Journal on Imaging Sciences*, 1(3):248–272, 2008. 43

J. Weickert et C. Schnorr : A theoretical framework for convex regularizers in pde-based computation of image motion. *International Journal of Computer Vision*, 45(3):245–264, 2001. 34

M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers et H. Bischof : Anisotropic Huber-L1 optical flow. *In Proceedings of the British Machine Vision Conference (BMVC)*, London, UK, September 2009. 12, 16, 43, 44, 48, 49

P. Wesseling : *An introduction to multigrid methods*. John Wiley and sons, 1991. 40, 92

J. Westerweel et F. Nieuwstadt : Performance tests on 3-dimensional velocity measurements with a two-camera digital particle-image velocimeter. *In Laser Anemometry-Advances and Applications 1991*, vol. 1, p. 349–355, 1991. 12

Y. Yamamoto et T. Uemura : Robust particle image velocimetry using gradient method with upstream difference and downstream difference. *Experiments in fluids*, 46(4):659–670, 2009. ISSN 0723-4864. 14

I. Yavneh et G. Dardyk : A multilevel nonlinear method. *SIAM journal on scientific computing*, 28(1):24–46, 2006. 40, 92

C. Zach, T. Pock et H. Bischof : A duality based approach for realtime tv-l1 optical flow. *In Pattern Recognition (Proc. DAGM)*, p. 214–223, Heidelberg, Germany, 2007. 8, 12, 16, 34, 43, 44, 48